MARY CAMPBELL

DOS 5.0 *At Work*

- Satisfaction Guaranteed – Money-Back Offer – See Back Cover
- Complete and Up-to-Date Coverage of DOS 5.0
- An Ideal Guide to DOS for All Beginning-to-Intermediate Users
- "Fact Finder" Tabs Allow You to Customize the Book

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial capital letters.

Library of Congress Cataloging-in-Publication Data

Campbell, Mary V.

DOS 5.0 at work / by Mary Campbell.

p. cm.

Includes index.

ISBN 0-201-57716-X

1. DOS 5.0. 2. Operating systems (Computers) I. Title.

QA76.76.O63C37 1990

005.4'46--dc20 90-23439

CIP

Copyright © 1991 by Mary Campbell

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America. Published simultaneously in Canada.

Managing Editor: Amorette Pedersen
Set in 11-point Garamond by Benchmark Productions

1 2 3 4 5 6 7 8 9-MW-9594939291
First Printing, February, 1991

To my husband, Dave

Table of Contents

Acknowledgments **x**

Introduction **xi**

The At Work Series xi
About This Book xi

New User Section: Getting Familiar with Your Computer System **xv**

Chapter 1: Getting Familiar with DOS Basics **1**

What You'll Learn 1
Defining What DOS Is and Does 2
Understanding the Main Functions of DOS 2
Understanding How DOS 5 Differs from Other DOS Releases 5
Understanding How DOS 5 and OS/2 Differ 7
Starting and Stopping DOS 8
Understanding the Options for Interfacing with DOS 5 10
Using the Mouse or Keyboard 17
Determining Which Version of DOS You Are Using 19
Clearing Your Screen 20
Setting the Date and Time 20
At a Glance 21

Chapter 2: Understanding File Basics **23**

What You'll Learn 23
Understanding What a File Is 24
Understanding File Types 26
Following File Naming Standards 29
Using Filename Extensions 30
Defining the Location of Files 33
Establishing Your Own Naming Conventions 34
Using Wildcards to Specify a Filename 36
Looking at the Way Files Are Stored 37
Obtaining a List of Files 38

Creating a Text File 43
Looking at the Contents of Text Files 44
At a Glance 45

Chapter 3: Typical Disk Configurations 47

What You'll Learn 47
Examining Disk Structure 49
Exploring Capacity Differences in Disks 52
Exploring Ways to Organize Information 55
Protecting the Contents of a Disk 57
Copying an Entire Removable Disk 59
Comparing the Contents of Two Disks 61
At a Glance 62

Chapter 4: Using the Shell for File Management Tasks 65

What You'll Learn 65
Loading the Shell and Exiting to the DOS Prompt 66
Exploring the Shell Screens 66
Interfacing with the Shell 70
Getting Help 75
Changing the View 77
Exploring the File Manager Screen 78
Selecting File Commands 82
Customizing the Shell 90
At a Glance 93

Chapter 5: Using the Shell to Work with Programs and Utilities 95

What You'll Learn 95
Working with Programs 96
Using the Utilities 110
At a Glance 114

Chapter 6: Managing Disk Information 115

What You'll Learn 115
Preparing Floppy Disks For Data Storage 116
Preparing a New Hard Disk 120
Copying a Disk 125
Comparing Disks 126
Working with Disk Labels 130

Checking the Status of a Disk 131
Moving and Replacing Files on a Disk 144
Renaming Disk Files 146
At a Glance 147

Chapter 7: Protecting Your Investment 149
What You'll Learn 149
Changing the Status of Files 149
Copying Files with COPY 152
Learning the Advantages of XCOPY 156
Creating a Backup 159
Using RESTORE to Recover from Data Loss 163
At a Glance 165

Chapter 8: Organizing Your Hard Disk for Productivity 167
What You'll Learn 167
Designing a Workable Structure for Your Data 168
Creating a New Subdirectory 173
Placing a New Directory Under the Root 173
Placing a New Directory Under the Current Directory 174
Placing a New Directory at Any Location 174
Changing the Current Directory 175
Removing a Directory 176
Looking at the Current Structure 177
Using a PATH Statement to Change the Search Path 177
Using APPEND to Modify the Search Path 181
Using JOIN to Use a Directory Instead of a Drive 182
Using SUBST to Shortcut Long Pathnames 184
At a Glance 185

Chapter 9: Using Your Printer with DOS 187
What You'll Learn 187
Using the PRINT Command 187
Printing a Screen 193
Sending Command Output to Your Printer 197
Using the MODE Command to Change Printer Configuration 197
At a Glance 201

Chapter 10: Reusing Your Command-Line Entries 203
What You'll Learn 203
Making the Most of Function Keys to Repeat Commands 203

Using DOSKEY to Maintain a History 208
Creating Macros with DOSKEY 211
At a Glance 215

Chapter 11: Working with the DOS Editor 217
What You'll Learn 217
Advantages Over EDLIN 217
Working with the Editor 218
At a Glance 236

Chapter 12: Defining a Series of Batch File Commands 237
What You'll Learn 237
Understanding What a Batch File Is and What it Offers 237
Entering DOS Commands in Batch Files 238
Executing a Batch File 242
Controlling the Display of Commands 242
Displaying the Status of ECHO 245
Interrupting Batch File Execution 246
Documenting Batch Files 249
Testing Batch Files 249
Looking at a Special Batch File 250
Creating a Few Sample Batch Files 252
At a Glance 256

Chapter 13: Adding Sophistication to Command Definitions 257
What You'll Learn 257
Using Parameters in Your Batch Files 258
Changing the Sequence of Execution 260
Linking Batch Procedures 271
Using CALL to Execute Other Batch Files 274
Repeating Batch Procedures 277
Using SET 279
Shifting Parameters 281
At a Glance 282

Chapter 14: Controlling Devices and Output 283
What You'll Learn 283
Understanding Device Names 284
Redirecting Input and Output 286
Using CTTY to Change the Default Input/Output Device 289

Using Filters 290
Building Pipes 297
At a Glance 297

Chapter 15: Customizing DOS 299
What You'll Learn 299
Changing the DOS Prompt 299
Changing CONFIG.SYS 302
Changing AUTOEXEC.BAT 310
Changing Drive Assignments 311
Adding Device Drivers 314
Changing the International Configuration 323
At a Glance 333

Chapter 16: Performance Tuning DOS 335
What You'll Learn 335
Before You Change Your AUTOEXEC.BAT and CONFIG.SYS Files 336
Improving Disk Efficiency 336
Improving Screen Speed 341
Improving Memory Utilization 341
Freeing Up Memory 345
Setting Up a Virtual Disk 347
Using Windows 349
Additional DOS Enhancements 349
At a Glance 351

Appendix A: Installing MS-DOS 5.0 353
DOS 5 Requirements 353
Upgrading Your System To MS-DOS 5 354
Installing MS-DOS 5 356
Changing Installation Settings for Your Monitor 356

Appendix B: DOS Error Messages 359

Appendix C: Key Summary 421

Appendix D: ASCII Codes 429

Appendix E: Sample Code Pages 433

Appendix F: DOS Commands 439

Index 531

Acknowledgments

Many individuals made major contributions in time, energy, and ideas to make this first At Work book possible. Among them are:

Chris Williams, editor of the At Work series, who worked on all facets of the project. Chris shaped the At Works series by adding new features, persevered with acquiring beta, worked on the marketing and promotion of the work, and insured that all the production on the book was handled flawlessly.

Amy Pedersen, Benchmark Productions, who handled all the production for the book. Amy did an excellent job of coordinating copy editor, art creation, and the production of final pages.

Steve Stansel, General Manager, Addison-Wesley, who supported the new series and worked closely with Chris to insure that we produced the best possible book in a timely fashion.

Julie Stillman, Acquisitions Editor, Addison-Wesley, who first worked with me on the DOS proposal and introduced me to Chris Williams.

Gabrielle Lawrence, who worked on all aspects of the project. Gabrielle did an excellent job of digging for answers on features that did not work correctly and helped in many ways to insure that this book was comprehensive.

Chris Raffo, who worked on several of the appendices and made sure that the error list was complete.

David Campbell, Jr., who put together some of the appendices and helped to check the final beta to insure that none of the command switches had changed.

Tanya Van Dam, Microsoft Corporate Communications, who made sure that we received all the beta releases in a timely fashion and was patient with our many questions about when we would receive final beta.

Christy Gersich, Microsoft Corporate Communications, who sent the final beta for DOS 5 and answered many of our questions.

Many individuals on the beta support staff at Microsoft who handled many last minute questions extremely quickly.

Introduction

Since its introduction in 1981, DOS has become the predominant operating system used on personal computers. There are over 60 million PC's that currently use DOS. The latest release, DOS 5 provides many new features that introduce new functionality and better support for the hardware sold today. These features include an optional DOS Shell interface that allows you to select files, change drives, and invoke DOS utilities with either a mouse or a menu selection. New utilities for unformatting and retrieving deleted files that formerly required the purchase of additional utilities are now part of DOS. Expanded memory management features that allows the use of memory above 640K, expanded hardware support that removes the limitation of two hard drives and two floppy disk drives, a full screen text editor, and an easy-to-use installation program.

The At Work Series

The At Work series is designed to meet the needs of both introductory and intermediate users. The books are designed to provide complete authoritative material on a particular software package concisely. The books in this series have a number of user-friendly features designed to make them easy to work with and learn from. Each book contains At Work sidebars to illustrate actual uses of the product in the real world. Adhesive tabs allow you to mark locations that you will want to refer to frequently. Chapter objectives and summaries help you focus on the important points that you will want to master in each chapter. A variety of appendices provide supporting details in any easy to reference section.

About This Book

DOS 5.0 at Work is designed to be the one book you will need for using all the features of DOS 5. The coverage is comprehensive ranging from basic DOS command entries or Shell selections to the creation of batch files and instructions for customizing DOS for your system. You will find examples of each feature within the chapters. The appendices include a quick command reference, a comprehensive list of error messages with suggestions for problem resolution, and other supportive materials.

You will find that the information in DOS At Work is organized for quick access. An outline of the contents of each chapter follows:

A section for new users that visually introduces the basic hardware components that DOS will manage. This section is especially useful for new computer uses since it explains the basics of the various components.

Chapter 1, Getting Familiar with DOS Basics, teaches you what DOS is and what it is used for. You will be able to try some basic DOS command entries in this chapter as you learn to clear your screen, check which version of DOS you are using, and start the DOS Shell.

Chapter 2, Understanding File Basics, introduces you to file basics such as filenames and types of information stored in files. You will learn to display a list of files and display the contents of a text file.

Chapter 3, Understanding Disk Basics, explains the structure of a disk and its capacity. You will learn to make a copy of a disk as well as how to compare two disks to see if they are identical.

Chapter 4, Using the Shell For File Management Tasks, teaches you to identify components of the Shell screen. You will learn how to make selections, perform file tasks, and customize the Shell.

Chapter 5, Using the Shell to Work With Programs and Utilities, teaches you how to work with one or more programs. You will learn about program groups and properties and how to use the Switcher.

Chapter 6, Managing Disk Information, teaches you how to work with large amounts of data efficiently. You will learn how to prepare disks for storage, create partitions and logical drives, duplicate disk data, and access deleted files or memory status information.

Chapter 7, Protecting Your Investment, gives you the tools you need to protect the data you create with your application programs. You will learn to run backup and restore, change the status of files, and copy your files.

Chapter 8, Organizing Your Hard Disk for Productivity, teaches you the techniques that you will need if you want to find your files quickly. You will learn how to select a structure for your hard disk to store them and you will learn to create, remove, and change directories.

Chapter 9, Using Your Printer with DOS, teaches you to set up your printer to work with DOS. You will learn how to print text files to your screen.

Chapter 10, Reusing Your Command Line Entries, teaches you to reuse your most recent DOS command entry to create additional commands. You will also learn how to use DOSKEY to work with a history of commands and how to create DOS macros.

Chapter 11, Working with the DOS Editor, shows you the basics of the EDLIN utility that allows you to edit lines in a text files. This chapter also introduces the new full-screen DOS Editor that allows you to edit text files using the full screen.

Chapter 12, Defining a Series of Batch File Commands, introduces you to the productivity that batch files can offer. You will learn how to display messages on the screen and how to wait for a user response. You will learn about the special AUTOEXEC.BAT batch file.

Chapter 13, Adding Sophistication to Command Definitions, allows you to create sophisticated batch files. You will learn how to use conditional logic and repetition in your batch files.

Chapter 14, Controlling Devices and Output, teaches you the various DOS device names. You will learn how to reroute command input and output.

Chapter 15, Customizing DOS, teaches you how to change the default DOS configuration. You will learn how to change the appearance of the DOS prompt, create a CONFIG.SYS file, change the international configuration, and include device drivers as part of the configuration.

Chapter 16, Performance Tuning DOS, shows you how to make changes that can improve the efficiency of DOS on your system. You will learn to minimize fragmentation, use extended, expanded, and reserved memory, and improve screen speed.

Appendices provide a DOS key summary, a quick command reference, a full list of error messages, ASCII codes, code page tables, and an installation appendix.

Why This Book Is for You

If you have used other releases of DOS you will find that this book is an easy-to-use reference to all of the new DOS 5 features. You will see the new features used in the context of meaningful examples and you will be able to apply them to your own situation. There is no need for you to proceed from chapter to chapter since each chapter as well as sections within each chapter stand alone and can be referenced individually.

If you are a new DOS user you will want to proceed through the chapters sequentially. You will find that this book moves at the correct pace and provides the learning aids that you will need to remember what you have read. The examples in the book are the ones that I have personally used to teach DOS to thousands of users in a classroom situation.

Getting Familiar with Your Computer System

If you are new to computers, as well as DOS, you will want to spend a few minutes exploring this section. You will learn to identify the various components of your computer called *hardware* through the many illustrations in this section. You will also learn the purpose of each hardware component. Once you have identified the various components of your computer you will learn about the instructions called *software* that tell your computer what tasks to perform.

Taking a Look at Computer Hardware

There are many components in your computer. Some components such as the keyboard are visible and can be touched easily. Other components are enclosed in a box and cannot be seen unless the cover is removed from the main computer unit.

Hardware components are made by a number of manufacturers. This helps account for the differences in size, speed, color, price, and capabilities of the same components on different systems.

Exploring the System Unit

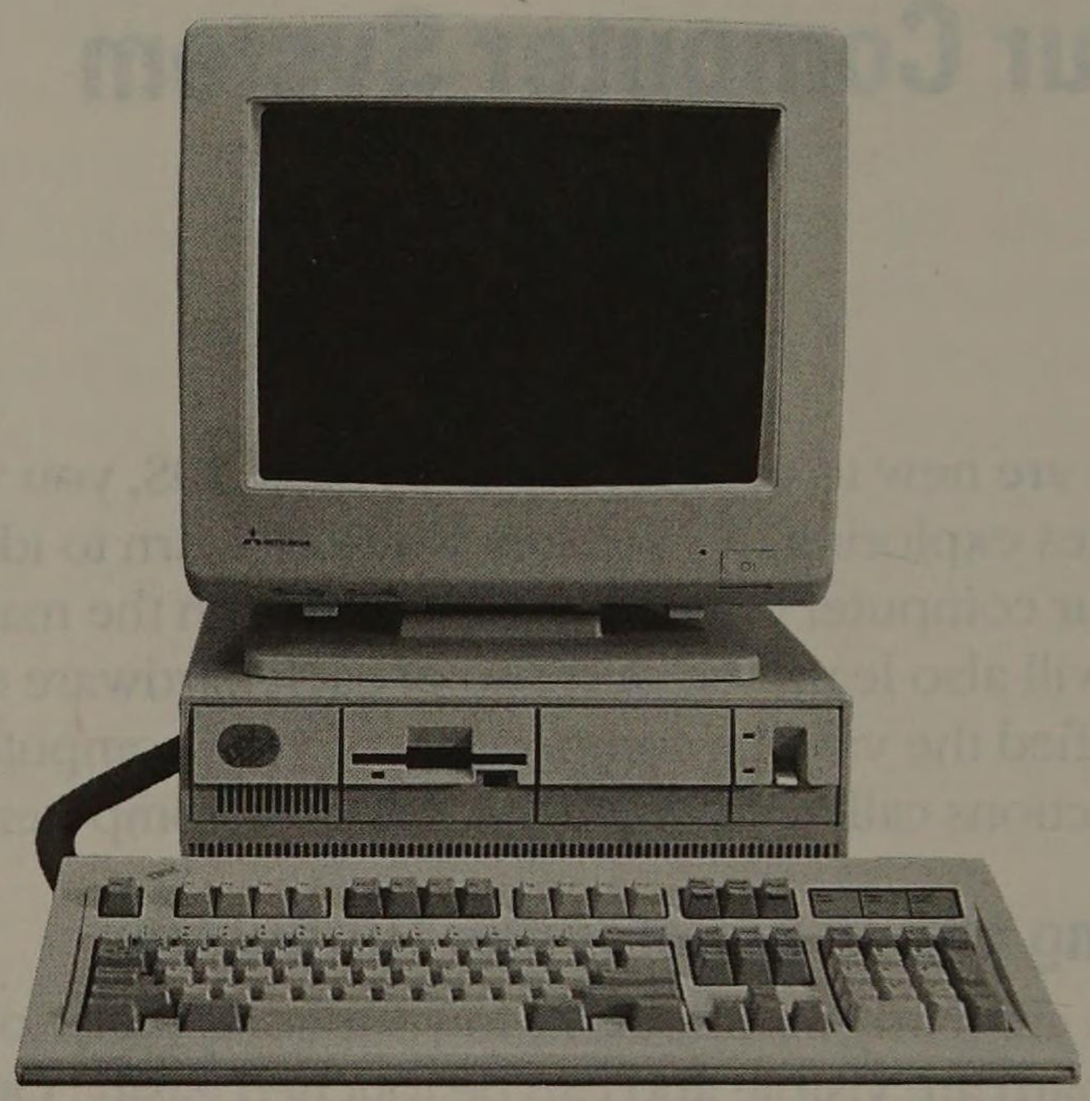
The system unit is the most important piece of hardware in your computer system. It contains the brains of your system. A component within the system unit is capable of remembering, at least for the short term. The system unit handles all the processing and computation that you need.

The system unit can take many shapes and sizes; in a desktop unit, it is typically a box on which the monitor or screen sits. Your unit might look something like the IBM PS/2 Model 70 desktop unit in Figure N-1. The *tower* models like the model 60 require less desktop space and are said therefore to have a smaller *footprint*. Portable units frequently have all the components attached directly to the system unit. A laptop or hand-held model would be smaller than any of the system units shown.

The system unit contains a number of components including memory, a central processing unit, slots for adapter cards, adapter cards, ports, and disk drives.

The system unit has an on/off switch to turn the power on to your system. This switch may be in the back of your system, on the side, or on the front. On most systems, the switch is pushed in one direction to turn the system on and in the other to turn it off.

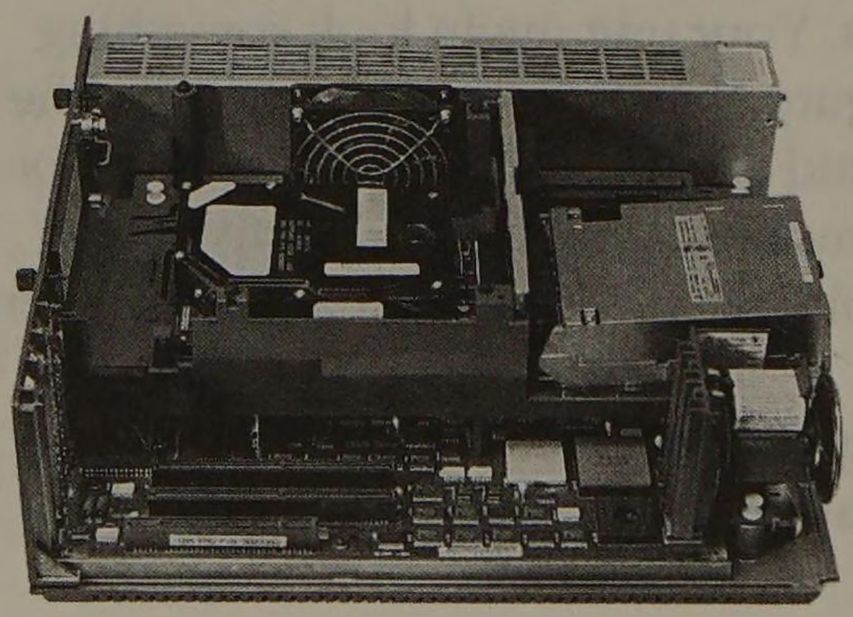
Figure N-1: IBM PS/2 Model 70



Looking Inside the System Unit The *motherboard* is the main circuit board in your computer. You must open the outer case of the system unit to see it. Figure N-2 provides a look at a motherboard from a popular computer system. Microprocessor chips are attached to this main circuit board. One type of chip applied to the motherboard is the processing unit, frequently called the Central Processing Unit (CPU). It might look something like Figure N-3. The CPU handles mathematical calculations and logical comparisons. Although the entire system unit is commonly thought of as the brains of your computer, it is actually this one component that provides most of this "brain power."

Memory chips are also located on the motherboard. These chips store information as long as the power to your computer is not interrupted. When you purchase a computer, the amount of memory in the unit is stated in either kilobytes (K) or megabytes (MB). A *kilobyte* of memory can store 1,024 characters, but rounding this number and thinking of it as 1 thousand is more common. A *megabyte* of memory can store 1,048,576 characters of information although it is common to think of 1 MB as 1 million. DOS, your application software, and your data will all use some of this memory as you are

Figure N-2: PS/2 Model 70 Motherboard



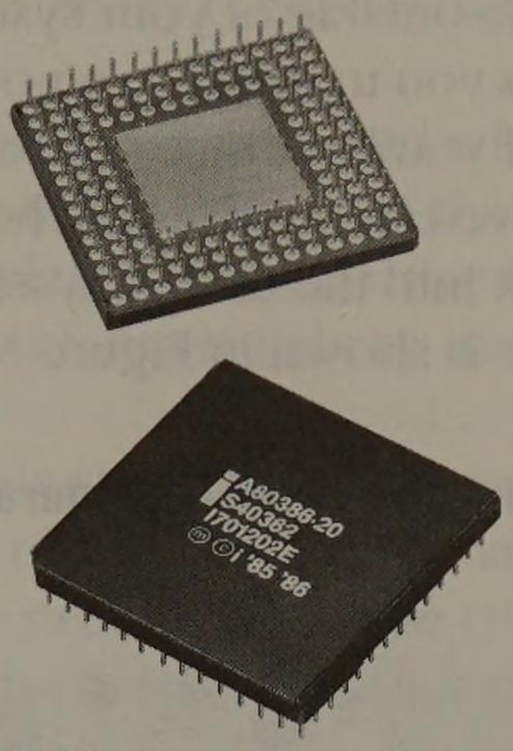
working with the computer. The memory in your computer system might reside in a bank of chips as shown in Figure N-2. You can also add additional memory that is stored on a card which in turn is placed on a card above the motherboard. This extra memory is either extended or expanded memory.

Slots on the motherboard allow you to add adapter cards. *Adapter cards* allow you to expand the basic capabilities of your system. They are placed in slots on the motherboard which are designed to allow users to define how they want to expand their system. One type of adapter card added to most systems is a video adapter card. This type of card defines whether your monitor displays color or black and white. The card also determines whether the monitor can display only text or text and graphics. Figure N-4 shows an adapter card that can be added to a system.

If your system has a hard disk, it will also be inside the system unit since it is not removable. The size of your hard disk is measured in megabytes with 40 MB and 60 MB the most popular sizes sold today. The information placed on the hard disk will be stored permanently, unlike the information stored in memory which is lost as soon as your system loses its power. Figure N-5 shows a hard disk with the circular plates providing the surface where information is stored. A hard disk is normally referred to as drive C.

Looking at the Outside of the System Unit If you look at the outside of the system unit you will see additional components. These are important—they include disk drives and

Figure N-3: Processing Unit



Courtesy Intel

Figure N-4: Adapter Card

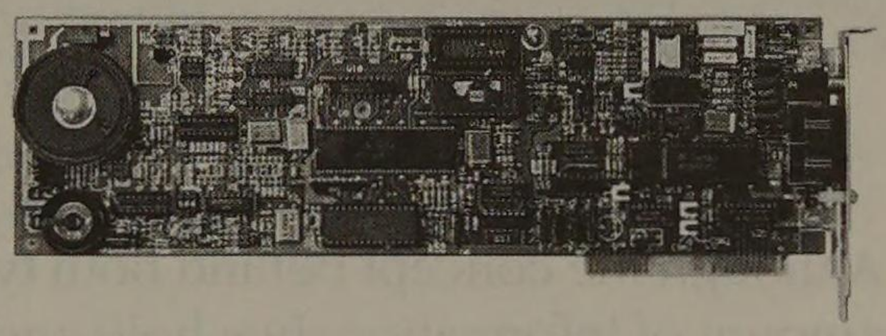
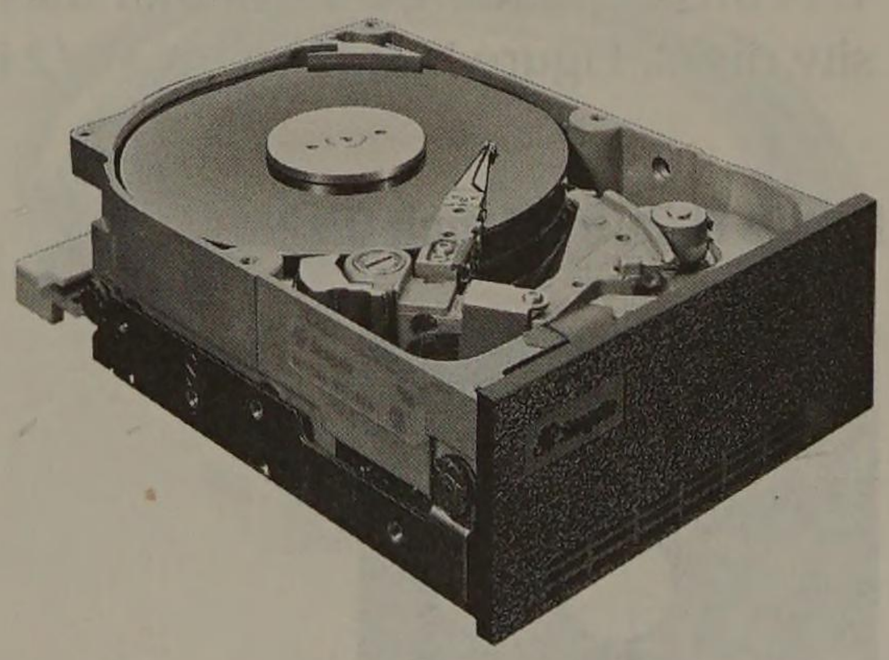


Figure N-5: Hard Disk Drive

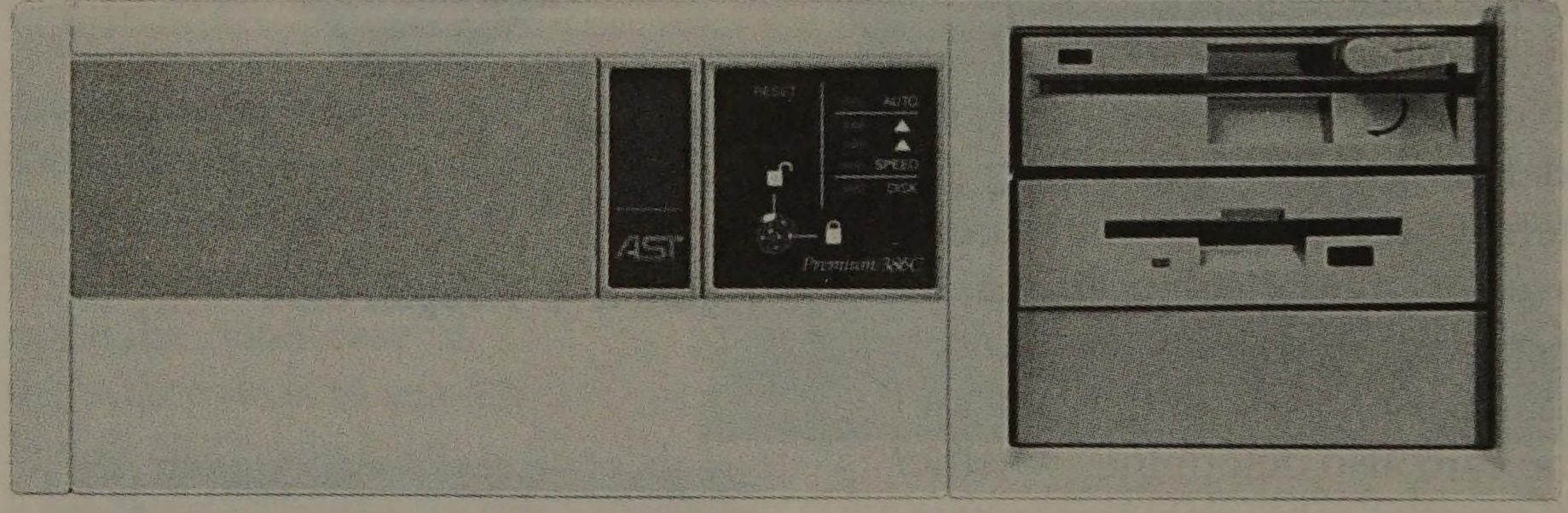


Courtesy Seagate Technology, Inc.

ports for connecting other equipment to your system.

On the outside of your system unit you are likely to see one or more drives that allow you to insert a removable diskette. The first one of these drives is called drive A. If you have a second drive as part of the system unit it is called drive B. The B drive can also be attached as an external drive if you do not have a second drive built into the unit. A system configured with a 3 1/2 inch drive and a 5 1/4 inch drive is shown in Figure N-6.

Figure N-6: Disk Drive Configuration



The most common diskette size in use today are 5 1/4 and 3 1/2 inches. Although the concept behind both types of disks are the same, they differ in the amount of information they hold and the protective covering over the actual disk. The 5 1/4 inch size can hold either 360K or 1.2 MB depending on the disks and the disk drive used to write the data. Figure N-7 shows a 5 1/4 inch disk. The 360K capacity disk is referred to as a double side/double density disk (DS/DD) and the 1.2 MB disk is a high density (HD) disk. The 3 1/2 inch disks hold either 720K or 1.44 MB depending on whether they are double sided, double density or high density disks. Figure N-8 shows a 3 1/2 inch disk.

Figure N-7: 5 1/4 Inch Disk

Figure N-8: 3 1/2 Inch Disk

High density (HD) disk drives can read either double side/double density disks or the high density variety. Double side/double density (DS/DD) disks can only read disks created on a double density drive.

You can write information to a disk as long as it has been prepared with the DOS formatting process. A 5 1/4 inch disk is placed into the drive with the notch on the left side of the disk and the oval cut-out edge placed into the drive first. A 3 1/2 inch disk is placed into the drive label face-up with the metal clip that protects the disk going into the drive first.

Once data has been written on the disk you can physically write-protect the disk to prevent the deletion of the data or the writing of additional information to the disk. On a 5 1/4 inch disk you need to use a write protect tab (these tabs are provided in most disk packages you purchase) over the notch on the left side of the disk. On a 3 1/2 inch disk you need to slide the small shutter toward the protective clip at the front of the disk.

The back of the system unit has ports that you can use to attach other devices to your computer. Figure N-9 provides a closeup look at the connectors for these ports. Cables like the one in Figure N-10 are used to connect the port to the external device. It is important that the cable connector exactly match the port connector. Devices you might attach to these ports include a keyboard, a monitor, printers, a modem, a mouse, or additional disk drives. These ports can be serial ports that accept a stream of characters that you can think of as marching single file from the port to another device or as parallel where parallel lines of characters proceed from the port to the other device.

Figure N-9: Port Connectors on the Back of the System Unit

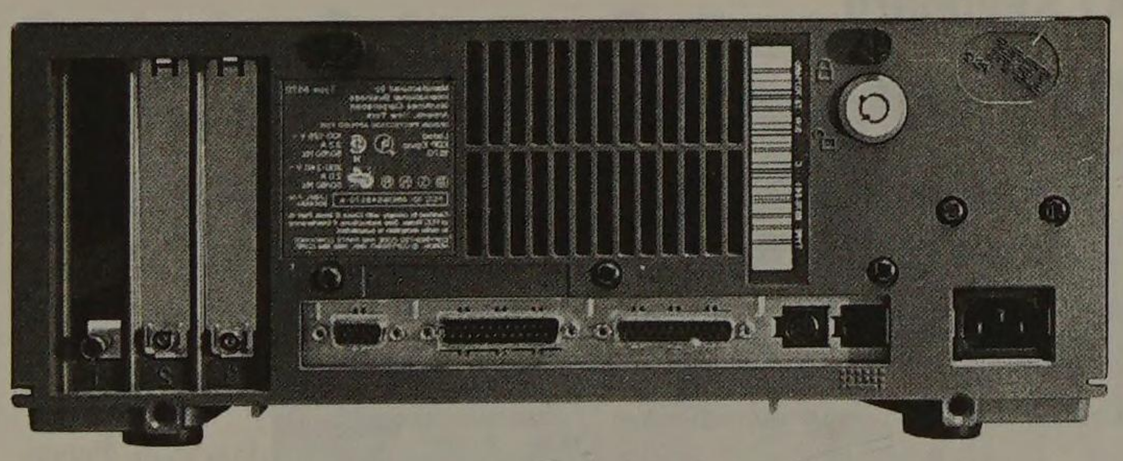
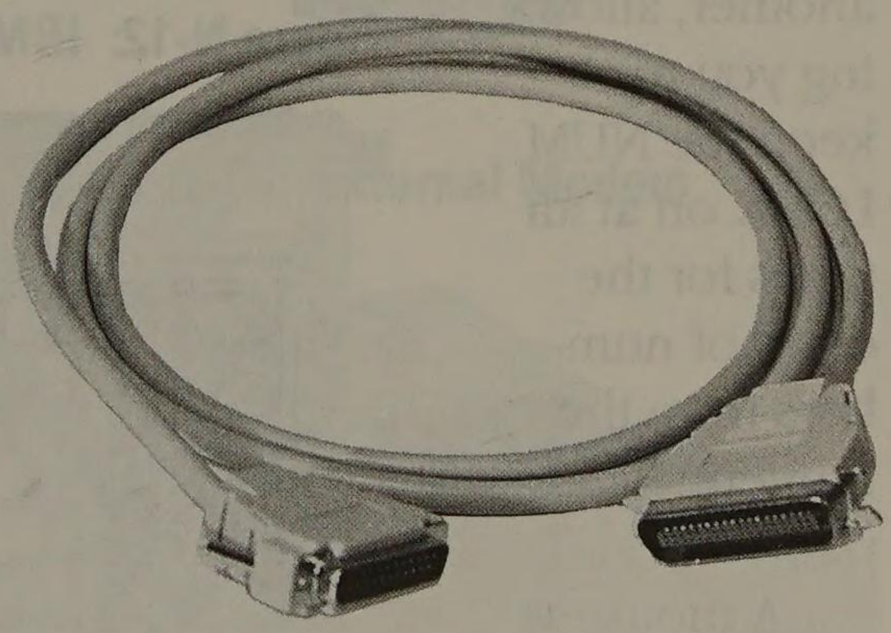


Figure N-10: Parallel Printer Connector Cable



Adding to the Basic System Features You need more than the system unit to have a usable computer system. At a minimum you need a device for communicating your needs to the computer and a way for seeing the output from your computer. A keyboard is the most common method of input and all computers have one. All keyboards have keys that correspond to typewriter keys. These keys always appear in the center area of your keyboard. Other keys appear in different locations on the various keyboards.

Function keys are assigned to request different tasks that vary by the application program you are working with. Function keys are the keys labeled F1 through either F10 or F12. They are located across the top of the IBM Enhanced keyboard shown in Figure N-11 or down the left side of the older model keyboard shown in Figure N-12.

The numeric keypad on the right side of the keyboard has dual functionality. With NUM LOCK on the keys on this pad enter numbers as you press them. Without NUM LOCK on these keys move the cursor from one location to another on your computer screen. Newer keyboards have separate keys to move from one location to another, allowing you to keep the NUM LOCK on at all times for the entry of numbers from the numeric keypad.

Figure N-11: IBM Enhanced Keyboard

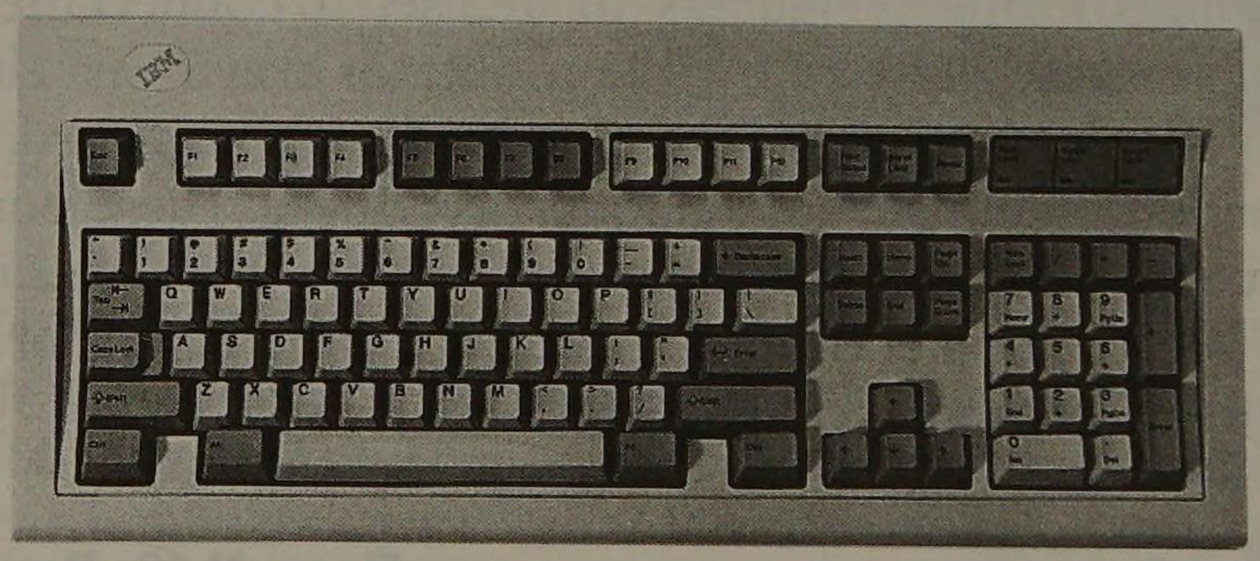
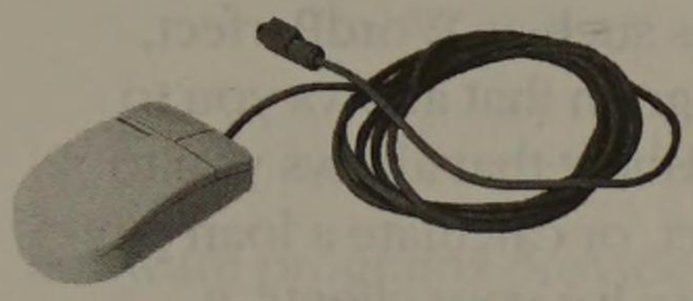


Figure N-12: IBM XT Keyboard

A mouse is an optional device that

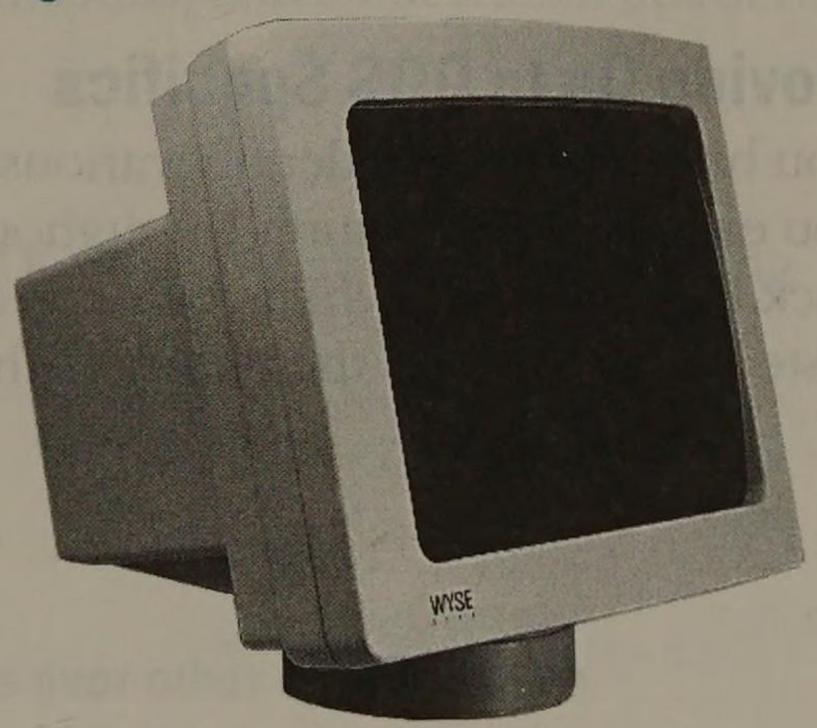
Figure N-13: IBM Mouse



you can use to limit keyboard entries. Figure N-13 shows a Microsoft mouse with two mouse buttons on the top. DOS uses only the left mouse button. Depending upon whether you press the button quickly and release it or hold the button down and move the mouse, you can select an item on the screen or an entire group of items. A mouse provides a shortcut approach to menu selections as well as a way of selecting a group of files in a list.

Monitor screens come in color or monochrome, which only displays one color. They can display only text characters like the ones you type on the keyboard or text and graphics images. Monitors differ greatly in resolution quality depending on the number of pixels in the displays. *Pixels* are dots that can take on an attribute. They are arranged in a grid shape and vary greatly from one type of monitor to another. The capabilities of your monitor are related to the video card installed in the system unit. Figure N-14 shows a monitor screen.

Figure N-14: Monitor Screen

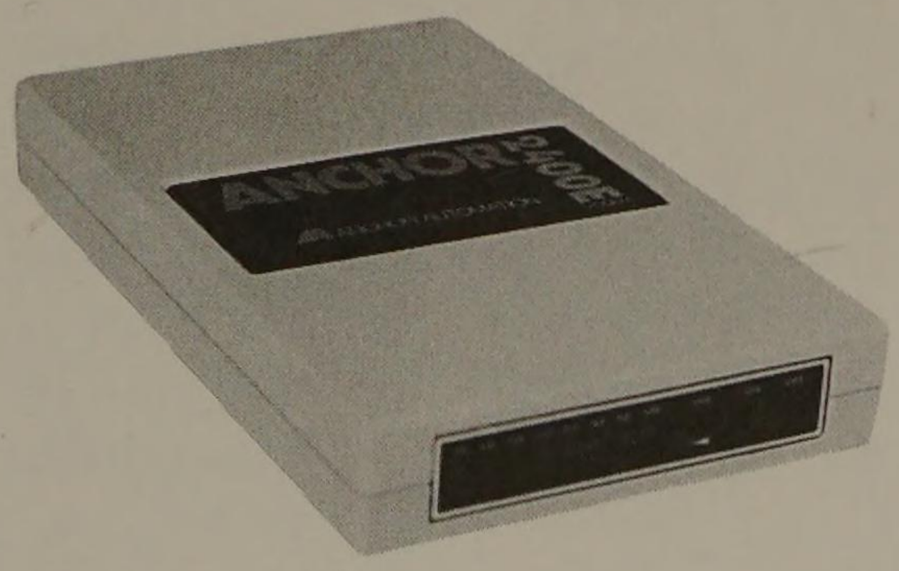


A *modem* is a device that allows you to connect your computer to other computers via the phone lines. You can use either an external modem or a micromodem which is a card inserted into one of your computer's expansion slots. Modems differ in the speed with which they can transmit and receive characters. Figure N-15 shows an external modem.

Exploring Software Options

Software consists of instructions for the computer that are stored as files on disk. There are different types of software. Operating system software such as DOS controls the resources of your computer and provides services to other software programs running on your machine. Operating system software must be the first software that is placed in your machine during each computer session.

Figure N-15: External Modem



Application software can be used to accomplish a specific business task. It can be general purpose or more specialized to meet the needs of a single industry segment.

General purpose software includes popular packages such as WordPerfect, 1-2-3, and dBASE. WordPerfect is a word processing program that allows you to type memos, letters, tables, and reports. 1-2-3 is a spreadsheet that allows you to create models to forecast sales, plan your monthly budget, or calculate a loan payment. dBASE allows you to create and maintain a database for your clients, a record collection, or inventory. You can use the features of these packages regardless of the business or industry segment that you work in.

Specialized software is tailored to one application. A specialized package might handle a task such as stock portfolio management, medical records, or legal docket control.

When you purchase an application software package you normally receive a manual and one or more disks. If you have a hard disk you will follow the install procedure provided with the software to copy the appropriate information to your hard disk.

Moving On to DOS Specifics

You have learned to identify various components of your computer system. As you explore DOS features throughout the chapters in this book, you can refer back to this section when you need a quick reminder of what components of your system look like and the purposes they serve.

Chapter 1

Getting Familiar with DOS Basics

Software programs provide the instructions that make your computer hardware perform needed tasks. Software programs include operating system software such as DOS that control the resources of your computer and application software such as WordPerfect and 1-2-3, which are suited to specific business tasks. Learning about DOS is an important step in mastering your use of the computer since DOS is the first computer program that you must run in your machine. Without DOS, you cannot use software such as WordPerfect, 1-2-3, and dBASE.

DOS 5 is the latest version of the DOS operating system software. DOS 5 provides exciting new features that were not present in earlier releases of DOS. With DOS 5 you will have better memory management capabilities, new commands, an easy-to-use menu system for selecting DOS commands, and a full screen editor for creating text files.

What You'll Learn

This chapter will provide an overview of some DOS basic features. After reading this chapter you will be able to:

- **Define what the DOS operating system is**
- **State the main functions DOS performs**
- **List some of the new features that DOS 5 offers over other DOS releases**
- **Start the DOS Shell and access its components**
- **Switch to the command prompt for entry of DOS commands**
- **Determine the version of DOS you are using**
- **Clear your screen**
- **Check the date and time settings**

Defining What DOS Is and Does

DOS is an *operating system.* Operating systems provide support services for other programs. This means that the operating system supplies general use routines and programs which handle hardware-related tasks for other programs.

The letters in DOS are an acronym for Disk Operating System since disks are the predominant method of storing data today. DOS is the most popular operating system for microcomputer systems. Other operating system alternatives include UNIX and OS/2.

DOS is a collection of programs provided as files. These programs provide services such as file and memory management and support for devices attached to your computer. They also interpret all DOS commands and handle both input and output activities on your computer. All programs that you run on your computer can use these DOS features. You can use DOS programs directly when you need to perform tasks involving disk or file management. Without the programs that DOS provides, you would need to interact directly with the hardware components of your system. This would limit the use of computers to programming experts capable of performing these complicated tasks themselves. Without DOS, application software options would also be considerably more limited since the development costs for a program would be much higher if every program had to provide the instructions for direct interaction with the hardware devices.

Although you may feel that you are interfacing directly with the hardware because you are typing at the keyboard, your input is translated and processed by DOS. It is DOS that communicates with the hardware. You are dependent on DOS for all of the tasks that are accomplished and for the messages that are directed back to your monitor screen.

Understanding the Main Functions of DOS

Many of the programs you use will have a single purpose. You can use a program such as WordPerfect to create a memo or use 1-2-3 to create a sales forecast model. DOS features are more varied and support every task you will ever accomplish with your computer. For example, you can use DOS to make a copy of your data, regardless of which program you used to create the data.

You will see an immediate effect on your computer screen as you perform some DOS commands. The DIR command displays the names of files on the current disk and directory on your screen. With other commands, your entries affect the contents of a disk but they do not display on your computer. The DOS DEL command deletes files from your disk but does not change the appearance of your screen when you are working from the command prompt. If you perform the

same delete action from the Shell (the easy to use DOS interface), the list of files will be updated to reflect the deletion.

Starting Your Computer

When you first turn your computer on each day, DOS performs the necessary housekeeping tasks to activate the DOS system in your computer. Part of this housekeeping is locating the hidden DOS files on your disk and loading the information from them into your computer's *RAM*, random access memory. These hidden files interact with your computer's *ROM*, read-only memory, and provide the basic input/output capabilities that DOS needs to provide disk and file management tasks for you.

Managing Hardware Resources

DOS is capable of interfacing with all the components of your computer system. DOS intercepts input to your computer system and interprets it. Input typed at the keyboard is the default. This input is processed by the DOS command interpreter stored in a file called COMMAND.COM. You can direct DOS to look for its input from a device other than the keyboard, such as a disk file. Programs that require data input from disk will make their requests to DOS and DOS will interact directly with the disk drive to obtain the needed data.

After DOS interprets your input, it processes the command by interfacing with all the needed components of your system. DOS uses the COMMAND.COM file to handle command processing. If DOS encounters any problems in carrying out a request, it returns a message to your screen or to the application program requesting DOS services.

If a request to DOS requires that output be returned, DOS handles the tasks of getting the information from the needed hardware device. This information might be a list of all the files on a disk or a copy of a file that you wish to write to another disk. DOS insures that the task is completed successfully and also notifies you or the requesting program of a successful completion.

Facilitating the Use of Other Software

Since specialized programming skills are needed to interface directly with hardware, software applications programs would be limited if DOS did not provide these features. The cost of developing proprietary solutions for commonly needed tasks would be prohibitive and discourage many of the start-up efforts that have produced popular business software applications.

If each software application interfaced directly with your hardware, a consistent approach would not exist for writing information to the disk. Without consistent rules for data storage, it is unlikely that files could be read by any program other than the program that created the file. It is also possible that some programs of poor quality could contain errors which would risk the integrity of data created by your other programs.

Putting You in Charge of Your Computer

Once you master DOS, your computer will no longer seem to have mystical power over your data and programs. You will understand what is happening when you save a file and how to control the actions of your computer system. You will learn how to start DOS as well as other programs. You will learn how to control the location of your data on a disk. With DOS commands you will be able to look at the contents of a hard disk and diskettes. You will have the tools you need with DOS to create protective backup copies of your data and will know how to restore your files if there is a catastrophic loss of data.

Although learning DOS will not make you an expert in computer hardware or architecture, it will provide a toolkit of skills for working with your computer. The skills in your toolkit will give you control over the data stored in your computer system. Since your data is your most valuable computer resource, you will realize a significant return on the investment you make in learning to use DOS more effectively.

Providing Tools to Enable You to Complete Tasks

DOS provides a set of utilities that allows you to manage the information stored on your computer. You can use DOS utilities to manage the information stored on your disk. You can also use them to accomplish tasks such as establishing an organization structure for the information on your disk, copying an entire disk of information, and preparing a new disk for storage.

You can use DOS utilities to work directly with individual files of information. DOS provides utilities for copying a file, renaming a file, printing a file, and deleting a file. These same utilities can be used to perform file management tasks, regardless of which program you used to create the file.

Although you must follow a specific set of rules to use the DOS utilities successfully, you will find that DOS provides a consistent set of rules which can be followed for all utilities. After you have mastered the syntax rules for the first few commands, you will find that you can apply these rules to all the other command entries. This consistent interface lowers the learning curve as you need to use new DOS features.

Understanding How DOS 5 Differs from Other DOS Releases

The first version of DOS was introduced in 1981. Just as application software programs have improved with each new release, so has DOS. Release numbers are represented by at least two digits. The first release of DOS was 1.0. A major new release is indicated by a change in the first number and a minor change by an increment in the second number. The changes from DOS 3.0 to DOS 4.0 are much more significant than the differences between DOS 3.2 and DOS 3.3. In almost every case the new release of DOS is downwardly compatible with prior releases. This means that files created with DOS 5.0 can be read on a computer with DOS 4.0 installed. There is a limit to this downward compatibility. The very early releases frequently are not compatible with today's new releases. As an example, DOS 5.0 is not compatible with DOS 1.0.

Some changes in DOS have resulted from improvements in computer hardware. The first version of DOS did not support directories on a hard disk because hard disk technology had not as yet been introduced for microcomputers. Although DOS 2.0 supported subdirectories for hard disks, it limited the size of the hard disk that could be used as a logical drive to 32 megabytes. The 32 megabyte limitation was not a problem when DOS 2.0 was introduced since hard disks had a capacity of only 10 megabytes. As disk capacities increased, the limitation was removed with DOS 4.0. The availability of expanded and extended memory has necessitated the addition of better memory management features in DOS 5. Also, now that most users are working with more than one program, DOS 5 offers multi-tasking. *Multi-tasking* allows you to have more than one program in memory at once and to switch between the different programs without ending the original program. This allows you to switch back and forth between the programs at will without the time delays that occur when you first load a program and the data you plan to use with it.

Other changes in DOS have resulted from a desire to improve the user interface to the operating system or to add new features requested by users. A menu interface known as the DOS Shell first appeared in DOS 4 and is also available in DOS 5. This menu interface allows for the selection of many different DOS commands without needing to use the exact syntax of a command prompt entry. Other interface changes include the addition of new commands or options for existing commands. DOS 5 offers an undelete command to restore deleted files and a new option for formatting disks that expedites the erasing of files from a formatted disk.

Examining the Many DOS Systems

This book focuses specifically on Microsoft's MS-DOS 5.0. In addition to the many versions of MS-DOS that have been introduced, there are product-specific versions of DOS that have a few subtle variations. These other versions of DOS are primarily provided by hardware vendors with their computers. If you buy a Zenith computer or a Compaq computer, a copy of Zenith DOS or Compaq DOS may be included with your machine.

Since the hardware vendors contracted with Microsoft to provide a specific version of DOS for their product, all of the offerings are very close to MS-DOS. Aside from the fact that earlier versions will not include the new DOS 5 features covered in the next section, the commands are almost identical. You will find an occasional difference in the spelling of a command or the ability to use a different parameter or option with a command.

Identifying the New Features

There are many exciting new features in DOS 5. These changes include behind the screen efficiencies in DOS's operation and new commands.

Better memory utilization is an important new feature of DOS. Now that many of the machines in offices have extended or expanded memory, DOS is poised to fully utilize this new memory by loading programs into high memory space. This leaves more room for your other programs in conventional memory. DOS 5 also requires less conventional memory than DOS 4 on older 8088 model machines. The MEM command first introduced in DOS 4 provides a much quicker way to check memory allocation than the CHKDSK option which was the only alternative in earlier releases. MEM also provides a more detailed look at memory allocation.

Multi-tasking capabilities in DOS 5 allow you to start multiple application programs and switch between them. This means that you can have a word processing program and a spreadsheet program in memory at the same time and switch between them at will.

DOS 5 has a new Shell which is faster and smaller than the DOS 4 Shell. The interface in this Shell also closely approximates the interface in Windows 3 for a more standardized user interface.

Help is available from both the DOS Shell and the command prompt. In the Shell, press F1 or use the menu option to invoke help. At the command prompt, type **HELP** and the command name. The last use of HELP produces the same result as when you type a command followed by **/?** to display help text on any DOS command.

DOSKEY is a utility provided with DOS 5. DOSKEY lets you maintain a history of all command prompt entries. You can use these entries again either individually or by combining them and saving them for later use. These saved keystrokes are called *macros* and are similar to macros that can be created for programs such as 1-2-3 or WordPerfect.

Command changes have extended the features of a number of DOS commands. Many new switches or options in the DIR command allow you to better control the display of information about files on a disk. You can display files with specific attributes, search subdirectories to locate a file, or indicate a sort order for your directory. An UNFORMAT command for hard disks allows you to undo an accidental formatting of your hard disk. Two new file attributes for the ATTRIB command allow you to set files as system files or hidden files. DOS 5 can support a hard disk with a partition larger than 32 MB without using SHARE.

Several command changes added with DOS 4 are still available in DOS 5. These changes include a prompt option when deleting files, the ability to rename directories from the Shell, and graphic support for the TREE command to look at a directory structure.

Quick Basic is now supplied with DOS in place of GW-Basic. Quick Basic is more powerful and provides online help.

A full screen editor makes it easy to enter and modify batch or text files. (Batch files are explained in Chapter 12.) Other data needed for testing purposes can also be entered into a text file with the editor. The ability to use its full screen capability makes changes easier since you do not need to list the program line by line. Its menu-based options are also an improvement over the single letter commands supported by its predecessor, EDLIN.

Benefitting from Compatibility with Earlier Releases

An improved installation program provided with DOS 5 can be used without a need to reformat your hard disk. This makes installation much faster and requires little interaction from the user. It eliminates the need for a backup and restore operation as part of your migration to a new version of the operating system.

Files created with DOS 2.0 and above are compatible with DOS 5. There is no need to rename files or make any other changes. You can use all of the new DOS 5 commands with files created under these earlier releases of DOS.

Understanding How DOS 5 and OS/2 Differ

There are many similarities between OS/2 and DOS 5. Many of the commands you use in DOS 5 provide the same features in OS/2. However, there are commands in

OS/2 that are not part of DOS 5. In addition, there are some differences in the requirements and interface for the two products.

Support for Older Model Machines

To run OS/2 you must have a newer model 286 or 386 machine. DOS 5 will run on older machines including the 8088. DOS works within the memory limitations of these machines and supports the use of the slower architecture they provide.

Supporting a Non-graphical Interface

Although the DOS Shell will provide a graphical Shell interface if you have a machine with a graphics adapter card, it will also run on machines without graphics support. This allows you to use all the new features of the product with text screen displays when you do not have graphics support.

Requiring Less Memory than OS/2

DOS 5 provides extremely efficient memory utilization. As with other DOS releases you can run the operating system on a system with a memory size as small as 256K and still have room to run a small application. If you have the more standard 640K system with as little as 64K of extended memory, you will have over 620K available for your applications. In machines with 1 MB, DOS can load TSR's device drivers, and application programs above 640K.

Even when working from the Shell you will find DOS 5 to be very efficient. Task swapping allows you to switch between multiple applications as they are swapped to disk for quick reaccess.

OS/2 extended edition requires a minimum of 2 MB for effective use. If you add OS/2 application software you are likely to need 2 to 3 MB of additional memory.

Starting and Stopping DOS

The procedure for starting DOS differs depending on whether you have a hard disk as part of your computer system. If you have a hard disk and follow the instructions in Appendix A, all of the essential DOS files will be copied to your hard disk as part of the installation process. You will not need to use the disk that came in your DOS package to start DOS. If you are using a floppy disk there is no way to store the essential DOS information in your computer, and you will need to use the DOS disk that you created with the floppy disk installation procedure in Appendix A.

Normally you will begin a DOS session at the beginning of each day when you first turn your system on. This session will continue to the end of the day when you turn your system off. DOS allows you to stop the current activity or to start a new session without turning your system off. You can use these options to completely clear the memory of your computer or deal with a situation when the hardware seems to be locked or repetitively processing the same information.

The Cold Boot Procedure To start DOS 5 on a hard disk system all you need to do is check to insure that drive A is empty and then turn your system on. This procedure is referred to as *booting your system.* This term bootstrap was coined to refer to the process of starting the computer each day. When you boot your system, DOS performs all the housekeeping tasks necessary to make DOS available. Part of this housekeeping is allowing the system to do a self-test when you power on. DOS then reads information from the boot disk contained in hidden DOS files. DOS checks for a CONFIG.SYS file on your boot disk and loads device drivers such as the one that makes the mouse available into memory. DOS also loads COMMAND.COM into memory to allow it to process DOS commands. If you have an AUTOEXEC.BAT file in the current directory, the DOS instructions in this file will be processed by COMMAND.COM.

The startup procedure just described is known as a *cold boot* since your system is off and cold when you begin the procedure. If you use the installation defaults, the next thing you will see is a blank DOS Shell screen. A box is superimposed on this screen titled **Reading Disk Information.** You can watch this box as DOS reports on the number of files and directories read from the disk. After DOS reads the entire disk, it fills in the DOS Shell window with information on the directory tree structure as well as individual filenames in the current directory. When you first boot a hard disk system, the files displayed will be from the root directory on drive C. When the Shell window is completed you are ready to make selections from the Shell or switch to the command prompt.

To start DOS on a system without a hard disk, place the DOS startup disk that you created with the instructions in Appendix A into drive A. Turn on your system and wait while DOS boots your system.

The Warm Boot Procedure Occasionally a program may encounter an error that locks your keyboard and other system resources. You may want to clear memory and begin again. Rather than turn your system off you can use a *warm boot* procedure that restarts DOS in your system. To perform this task press CTRL-ALT-DEL. If this key combination looks a little strange to you, it was selected since it is an unlikely

combination to press accidentally. If you do not have DOS on a hard disk, you will need to put the Startup disk in drive A.

- ***NOTE:*** Both the cold boot and warm boot procedures clear memory. This means that they remove any application software and data from memory. Although you will have the software on a disk, if the data is not saved on disk, it will be lost when you perform this procedure.

Stopping the Current Task Sometimes you will want to interrupt the current DOS tasks rather than restart DOS. Interrupting the current task is much less disruptive than restarting the system since all the information you have in the memory of your machine will not be lost. You can press CTRL-BREAK to stop the current DOS task. The word BREAK is usually located on the key that also reads Pause or Scroll Lock. CTRL-C is another option for stopping the current DOS command but it is effective in fewer instances than CTRL-BREAK.

Understanding the Options for Interfacing with DOS 5

When you start the DOS Shell by typing **DOSSHELL** and pressing ENTER, the initial DOS screen you see provides a graphical display of the various DOS elements in what is known as the DOS Shell. If you are using a floppy disk system, you will want to put the disk labeled Shell in drive A before starting the Shell. You can perform many tasks easily with menu selections from this display. If you prefer, you can also work directly with the DOS command prompt for a fuller range of options.

Using the Shell

The DOS Shell can provide a quick start approach to DOS tasks. Since it is likely that you are already using another program such as 1-2-3, an accounting system, or WordPerfect, you will already have some familiarity with making menu choices. DOS uses standard interface options for menus. The pull-down menu approach allows you to view all the options. The accelerator keys listed to the right of some menu selections provide a short cut approach to command selection without activating the menu first. You can use a mouse to directly click the selection you desire or, if you are using the keyboard, you can activate the menu and then type the underlined letter in the menu selection.

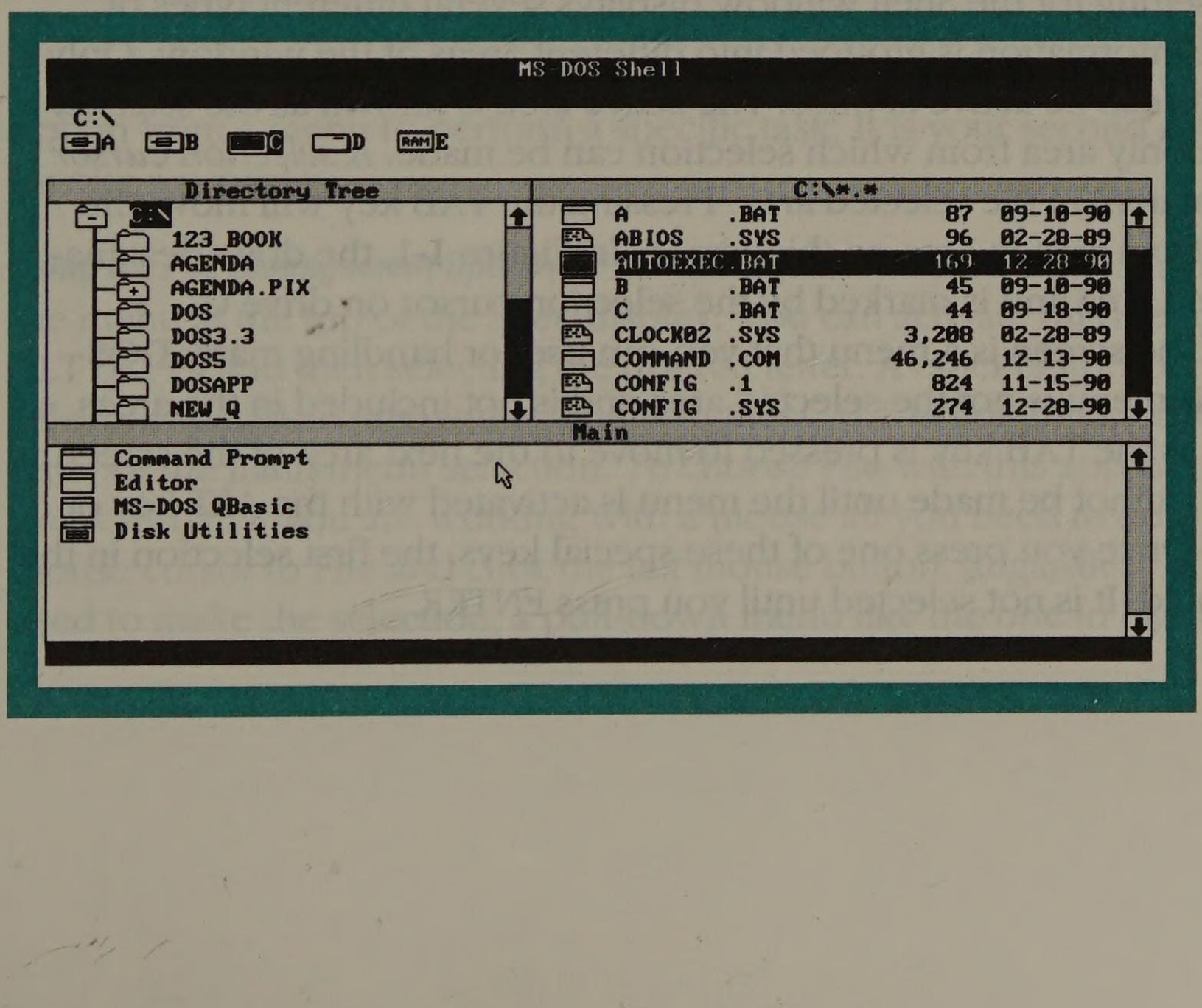
Another option is to select the menu option by pressing ALT or F10 to activate the menubar, then typing the letter that begins the menu selection and making the selection from the pull-down menu. To select File New with this approach you

would press ALT-F then type **N**. Regardless of the method you choose, information about what you need to enter is always displayed on your screen when you use the Shell. This eliminates the guesswork in correct command entry. Also, dialog boxes that appear after making a selection allow quick selections and entries when additional information is needed.

Switching Back to the DOS Shell If you are working at the command prompt you will not want to reboot your system to activate the shell. You can type **DOSSHELL** at the command prompt to start the DOS Shell. When you are ready to return to the command prompt, you can press F3 or ALT-F4 to leave the Shell permanently. If you only want to enter a command or two, press SHIFT-F9 and the command prompt will also appear. The difference is that with SHIFT-F9, you can type **EXIT** to return. DOS will quickly load the Shell since it saved the information formerly displayed on your screen.

Understanding the Shell Window If your monitor supports graphics, the DOS Shell will display a number of graphic elements for the information represented in the Shell window. Although you can choose to display only a list of programs on the Shell window, the combined file/program list is the default display and the most useful since it provides access to both file tasks and programs. With a graphics monitor, your screen will look like Figure 1-1 with the file/program list displayed.

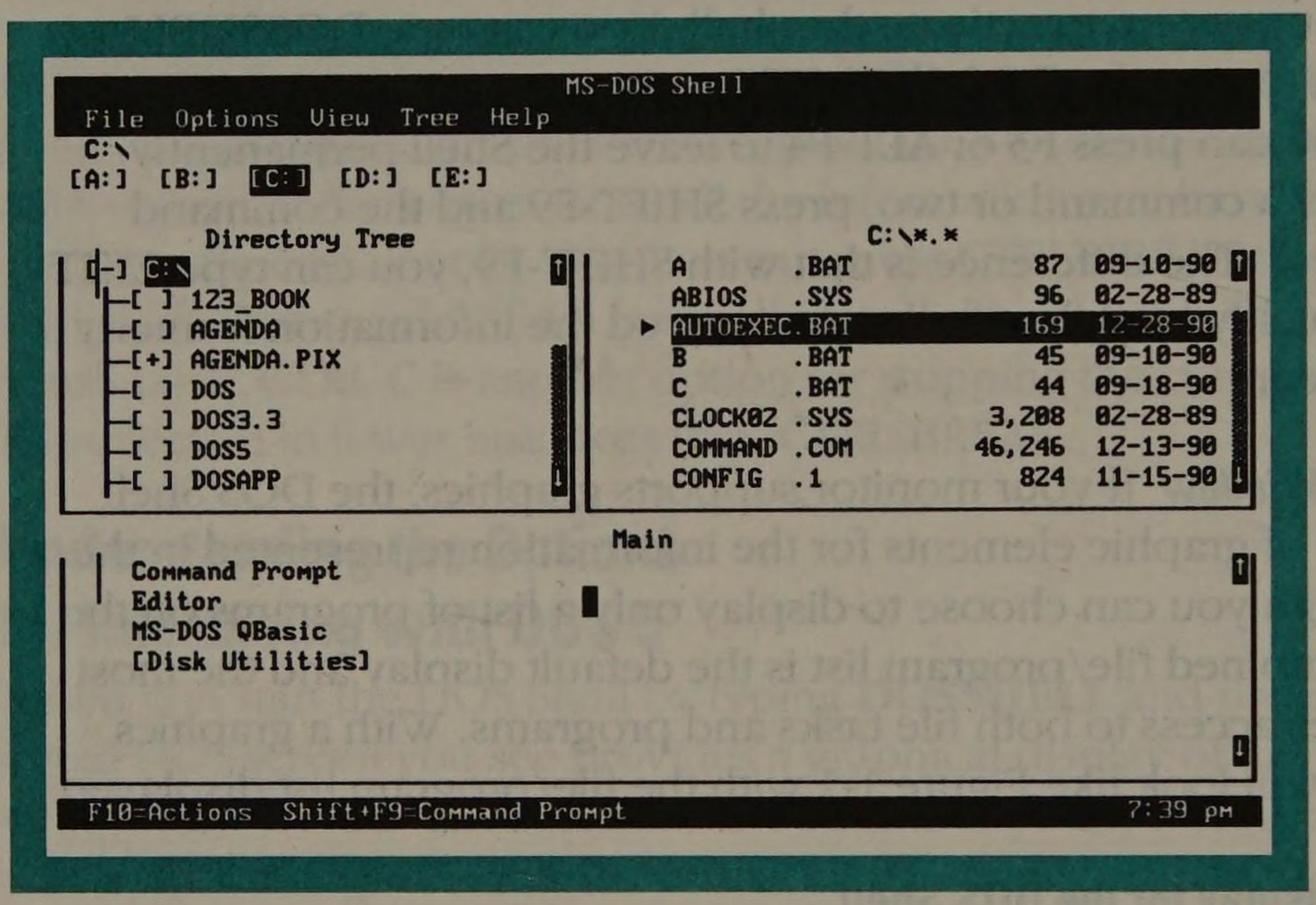
Figure 1-1: Graphics Display for the DOS Shell



The different sections of the screen display the drives, directories, files, and programs you can select.

The arrow on the screen is the mouse pointer that marks the current position of the mouse on the screen when a mouse is installed. The graphic elements are replaced by text characters when you screen does not support graphics and will closely match Figure 1-2.

Figure 1-2: Text Display for the DOS Shell



The default setting for the Shell window displays several different types of information. The information is grouped into different areas of the window. Only one of these areas can be active at once. The active area is known as the *selected area* and it is the only area from which selection can be made. A *selection cursor* marks the current item in the selected area. Pressing the TAB key will move the selection cursor from area to area on this screen. In Figure 1-1, the drive designation is the selected area and is marked by the selection cursor on drive C.

At the top of the screen is a menu that you can use for handling many DOS tasks. Initially this menu is not the selected area and is not included in the areas that are affected as the TAB key is pressed to move to the next area of the screen. Menu selections cannot be made until the menu is activated with the ALT key or by pressing F10. Once you press one of these special keys, the first selection in the menu is highlighted. It is not selected until you press ENTER.

If you have a mouse you can activate the menu and make your selection from the list of menu options in one step. All you need to do is move the mouse pointer to the desired menu selection and click the left mouse button.

Beneath the menu are graphic elements that you can use to select the active disk drive. When the Shell is initially loaded, this area is active to allow you to change the drive selection if you wish. The current drive is highlighted by the selection cursor.

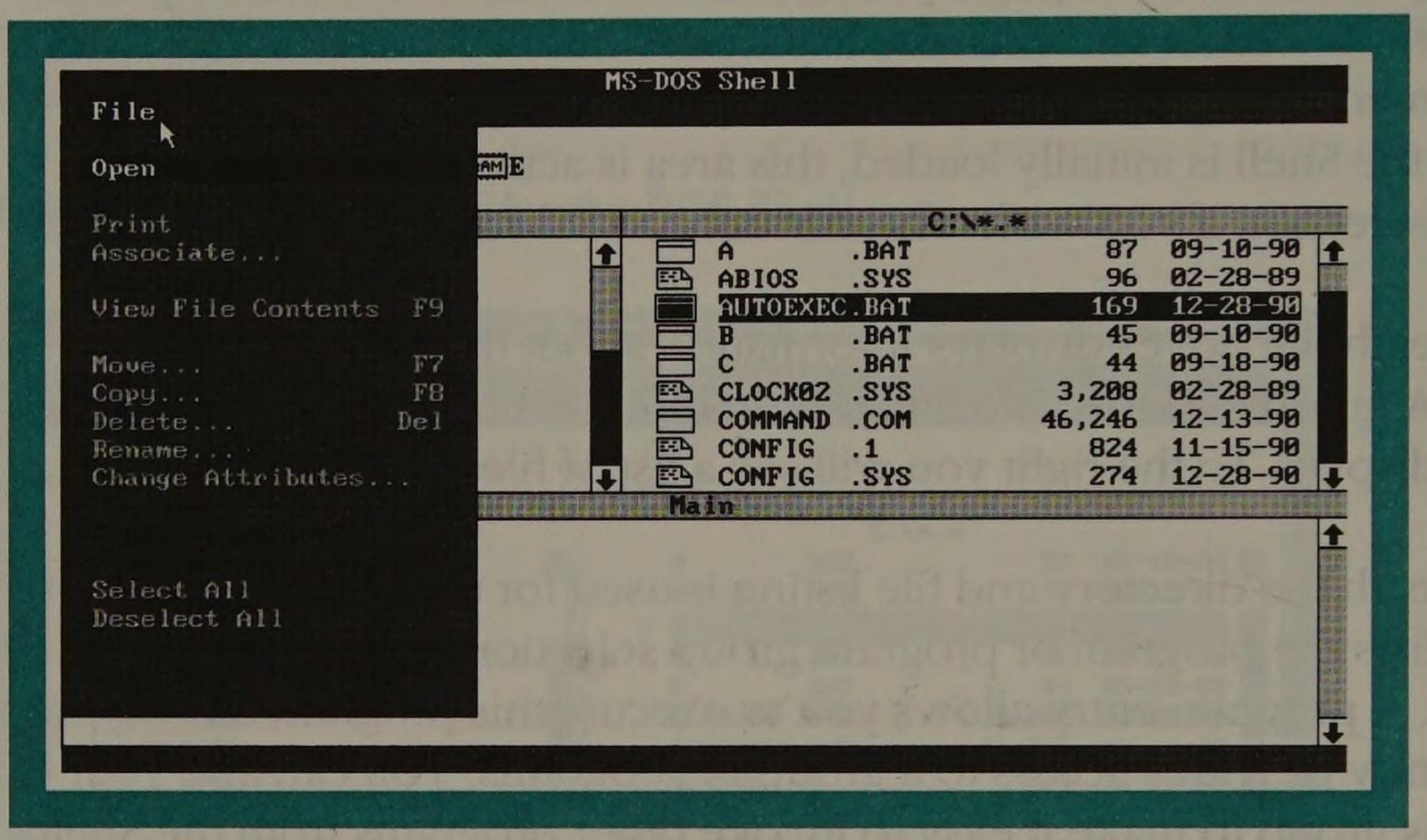
The area beneath the drives displays information about the directories and files on the disk. A graphic tree of the directories is shown on the left if your monitor supports this display. To the right you will see a list of files present in the initial directory.

The area beneath the directory and file listing is used for the main program group. This area lists the program or program group selections that are available through the Shell. A program entry allows you to execute this program directly. A program group entry provides access to a group of programs. You can add your own entries to this group to make it easy to invoke other programs from the Shell.

The initial program selections in the Shell provide a few basic options that you will use frequently. Command Prompt is a selection that will place you at the DOS prompt. Editor provides a full screen editor for the creation of DOS batch files or text data files. Although the EDLIN utility from earlier DOS releases is still available, the new editor is much easier to use. When you select MS-DOS QBasic you can create a program with Quick Basic. The Disk Utilities are a collection of DOS disk maintenance tasks that you can perform with menu selections. Unlike the other entries, Disk Utilities are a program group entry. Once selected you will see a list of the specific programs that are part of the group and you can select an option from a menu to perform a specific task. It is your second selection that will invoke a program.

Using the File Management Options File management options are available through the menu at the top of the Shell display. You can activate this menu by pressing ALT or F10 and then selecting the desired letter. A quicker approach which also selects a specific command option is to press ALT in combination with the first letter of the main menu selection. To choose File with this approach you simply press ALT-F. If you are working with a mouse all you need to do is move the mouse cursor to File and click the left mouse button. Regardless of the procedure used to make the selection, a pull-down menu like the one in Figure 1-3 appears.

Figure 1-3: Pull Down File Menu



One letter of each menu selection is boldfaced or underlined to indicate the letter you can type to select it.

Using the Main Program Group The options in the main program group provide access to the command prompts, the Quick Basic programming language, a full screen editor, and several disk utilities. In Chapter 6 you will learn to create additional entries for your own programs and groups of programs.

When you want to work in the main program group you must make this area active. When the drive selection is the active area you can press TAB three times to select the program group area. A shortcut is pressing SHIFT-TAB since it will move the selected area in the opposite direction. In this situation it would move the selection cursor from the first area for the drive selection to the last area, main program group.

The Disk Utilities within this group provide a quick way to format a disk, backup or restore a hard disk, and copy a disk. You can complete any of these tasks by selecting Command Prompt rather than Disk Utilities except that you will need to do more typing. When you use the command prompt you also need to know the exact syntax of every command you use. The Disk Utilities selection saves time since it does most of the typing for you.

If you accidentally select one of the Main program options, press ESC. ESC cancels your selection and returns you to the Main program list.

Moving Between Components Moving between components of the DOS Shell window is easy. Each time you press TAB, the next area of the screen is activated. When you press SHIFT-TAB the selection moves to the previous area.

You can also move outside the Shell to the DOS prompt. There are several ways of achieving this. One option is to select Command Prompt from the main program area of the window at the bottom of the screen. You can press the TAB key until this area of the screen is active and press ENTER to select Command Prompt. Another option is clicking the mouse on the desired selection. Other options for moving to the command prompt are pressing ALT-F4 to permanently remove the DOS Shell from memory or to press SHIFT-F9 for a temporary exit. If you use the SHIFT-F9 approach, DOS saves the Shell settings. You can restore the Shell by typing Exit.

Using the Command Prompt

If you want to access all the commands and options DOS has to offer you will want to use the command prompt method of invoking DOS features. When you use the command prompt, your requests are entered with commands. *Commands* specify to DOS the exact action you wish to take as well as any special information needed to carry out your request.

Determining the Active Drive When you use command prompt entries, a DOS prompt appears to indicate that DOS is ready to process your next entry. The initial setting for this prompt displays a letter representing the active drive.

The drive letter in the DOS prompt is followed by a greater than symbol (>). This symbol serves as a separator character between the DOS prompt and the entries that you make.

If you have more than one drive you can change the active drive by typing the drive letter followed by a colon. For example, to change from drive C to drive A, type:

```
A:
```

As long as DOS finds a readable disk in drive A, DOS will activate this drive. If the drive you specify is empty DOS will display the message:

```
Not ready reading drive A
Abort, Retry, Fail?
```

You can insert a disk then type an R for retry. If you type F for Fail, DOS will display the message:

```
Current drive is no longer valid>
```

You can then enter a new drive specification.

Command Syntax Every DOS command consists of one to three parts. The first part, the command name, is always required. A *command name* can be a word such as DATE or an abbreviation for a word such as VER, representing version. The second component, an object or *parameter* is optional. In an entry such as DIR A:, DIR is the command which requests a directory listing. The optional parameter, A: tells DOS you are interested in the directory of the disk in drive A. *Switches* are the third component and are also optional. Switches can alter the actions of the command. If you elect to use switches you will find that they always begin with a slash (/) and are followed by a single letter. Switches are unique to a specific command. Many switches can be used in combination with other switches for the command. An entry of DIR A: /p tells DOS to pause after listing the first screen of directory entries. Without this switch, a directory containing more file entries than would fit on a screen would cause information to scroll down the screen faster than you could read it. Adding the /p switch allows you to control when additional information is displayed.

When entering your command, if you make a mistake, which invalidates your request, DOS will display an error message. The error message will depend on the type of mistake you make in the entry. Table 1-1 shows some of the most common error messages encountered.

Table 1-1: Common DOS Error Messages for Syntax Errors

Message	Meaning
File not found	Filename specified incorrectly
Bad command or filename	Command misspelled or space omitted before parameter
Invalid switch	Switch specified is not used with the current command
Required parameter missing	Command requires a parameter but none was supplied
Invalid drive specification	Drive specified is not valid on the system

Using Both Internal and External Commands Some DOS commands are available as soon as you load DOS. Other commands require DOS to execute a program in order to perform the requested feature.

Commands which DOS can support without executing a program are called *internal commands*. As soon as DOS is loaded in your machine, these commands are immediately available. You will learn to use several internal commands in this

chapter as you try DATE, TIME, and CLS. In later chapters you will try other internal commands that check the contents of a disk or copy files.

Commands which require DOS to locate and execute a separate program are called *external commands.* DOS will search the current disk drive and directory for the program name indicated by the command. If you want DOS to look in other locations for the file you will need to use the PATH command discussed in Chapter 9 to direct DOS to other directories. If DOS cannot locate the needed program, it will display the message:

```
Bad command or file name.
```

Changing the Interface Used

If you have a monitor that does not support graphics, you can still use the DOS Shell. The install procedure selects a text screen mode. But if your hardware supports it, you can use a graphics screen. In text screen modes, all of the graphic icons will be replaced by text, but the features will be the same.

Using the Mouse or Keyboard

You can use either the mouse or the keyboard to make selections from the DOS Shell. The same commands are available with each approach although the mouse often offers a significant time savings since you can directly select the desired command rather than having to make a series of menu selections to finalize your choices.

Using the Mouse in the Shell

Even though you have a mouse attached to your computer you may find that a mouse cursor does not appear on your screen. The absence of a mouse cursor indicates that your mouse has not been installed and that it is not in operation. To install your mouse you will need to modify your CONFIG.SYS file. You will find the instructions you need to make this modification in Chapter 15.

Once your mouse is installed you can use it to make all of your selections from within the Shell. You will need to master three different techniques to use the mouse effectively. This are clicking, double-clicking, and dragging. *Clicking* means moving the mouse cursor to the desired object and pressing the left mouse button once. You click a menu item to select it. *Double-clicking* requires you to move the mouse cursor to the desired item and press the left mouse button twice in rapid succession. You can double-click a drive to have DOS reread it. *Dragging* requires you to move the mouse cursor to the first item or area that you are interested in. Once the mouse pointer is positioned you can depress the left mouse button and hold it down while you move the mouse pointer to the ending location.

These three basic techniques will be used for everything from changing the active disk drive to selecting files and making menu selections.

Using Special Keys

DOS supports many time-saving keys and key combinations. These key assignments vary from the Shell to DOS Prompt entries. Even within the Shell you will find that keys are assigned to different functions in the areas of the Shell window. In this section you will learn about the general-use Shell keys as well as ones that allow you to move within the Shell or between the Shell and DOS Prompt. These keys are summarized for quick reference in Table 1-2.

Table 1-2: Key Assignments for the DOS Shell

Key	Action
ALT	Activates the menu
ALT-F4 or F3	Exits the Shell and displays DOS Prompt
ARROW (UP or DOWN)	Moves up or down one line
CTRL+END	Moves to the bottom of a list
CTRL+HOME	Moves to the top of a list
END	Moves to the end of a list
ENTER	Selects the current item
ESC	Cancels requests
F1	Requests help for the current activity
F10	Activates the menu
HOME	Moves to the beginning of a list
letter	Moves to the next item that begins with the letter you typed
PgDn	Moves to the next window of information
PgUp	Moves to the previous window of information
SHIFT-F9	Saves the current Shell settings and goes to the DOS command prompt so you can return by typing EXIT.
SHIFT-TAB	Moves in a counter-clockwise direction to the previous area of the DOS Shell window
TAB	Moves in a clockwise direction to the next area of the DOS Shell screen

The F1 key is used as an alternative to the Help menu. Using F1 can offer an advantage since it provides context sensitive help which is tailored to your current activity. When you use the menu option for Help you will need to select a help topic from the menu. ESC is another key used throughout the Shell. ESC always indicates that you want to cancel the current activity and return to your previous status.

You have already learned about the combination keys that exit from the Shell. Pressing F3 or ALT-F4 exits from the Shell permanently while SHIFT-F9 saves the Shell settings for a quick return. Another option covered earlier in the chapter is the use of the TAB key to move to the next area on the screen proceeding in a clockwise fashion. To move to the previous area proceeding in a counter-clockwise direction, press SHIFT-TAB.

You can move within any list on the screen in a number of ways depending on how quickly you want to scroll through the options. Pressing HOME will move you to the top of a list or the beginning of a line. Pressing END moves you to the end of a line or list. You can use CTRL in combination with HOME or END to move to the beginning or end of a list but these extra keys only require more work. PgUp moves you to the previous window of information and PgDn moves you to the next window. The UP and DOWN ARROW move you a line at a time in the direction of the arrow. Typing a letter moves you to the next item in the list that begins with the letter. If the selection cursor is on an entry that begins with A and you want to move to an item that begins with S typing an S is the quickest and most precise way to reposition the selection cursor.

You can invoke the Shell menu with either ALT or F10. Invoking the menu does not select a command. You must do that by highlighting the command and pressing ENTER or by typing the indicated letter.

Determining Which Version of DOS You Are Using

With so many different versions of DOS still in use through business operations, you will want a way to determine which release of DOS is available on a specific system. You can use the VER command to determine exactly which release of DOS is installed on your system. When you type **VER** at the command prompt DOS displays a message that looks like this:

```
MS-DOS Version 5.00
```

Clearing Your Screen

As you enter DOS commands at the command line, DOS continues to use the next line on your screen for the entry and the display of any results or messages. When all the screen lines are used, DOS begins to scroll lines off at the top of the screen to make room for the new entries at the bottom. You can clear all the entries from your screen to allow new commands and messages to begin displaying at the top of your screen.

To clear all the entries from your screen, type **CLS** at the DOS command prompt and press ENTER.

■ ***NOTE:*** You will want to remember the CLS command later when you begin to build custom DOS commands with batch files. It allows you to clear a screen of all extraneous entries, then display a menu or other information to which you want to direct the user's attention.

Setting the Date and Time

You can check or set the system date and time whenever you have the DOS prompt displayed on your screen. This ability allows you to correct these settings before you start a program that will execute the program or to use the temporary DOS exit that many programs now support to make the change. Your entries must provide valid data and time components. For example, you cannot specify a month of 13 or a day of 30 if you tell DOS that the month is February.

■ ***NOTE:*** There is an easy way to exit to DOS with a quick return to your application using most popular software packages. To use the temporary DOS exit in 1-2-3, type **/S** to invoke the System command. To exit to DOS in WordPerfect, press CTRL-F1 then type **1**. To exit to DOS in dBASE IV, use the Tools menu to select Dos Utilities then press ALT-D to select DOS followed by **G** to select Go to DOS. You can return to your application and the model, worksheet, or database that was active before DOS when you type **EXIT** and press ENTER.

Displaying and Setting the Date from the DOS Prompt

To display the current date setting, type **DATE** at the DOS prompt and then press ENTER. DOS displays the date in the following format:

```
Current date is Sat 12-15-1990
Enter new date (mm-dd-yy)
```

To change the date type a new date in the format mm-dd-yy or mm/dd/yy. To enter the date for October 20, 1991 you would type **10-20-91** and press ENTER. DOS changes the date in the system clock for the current session in all computers. If you have a newer computer with the clock on the motherboard, changing the date or time will alter these settings for all subsequent DOS sessions. If your system has a separate clock card, you may need to use the special software provided when you bought the clock card to change the date and time permanently.

- ***NOTE:*** If you do not have a clock card in your system, you will want to enter the date and time each time you start a new DOS session. If you do not enter the date and time, you will not be able to distinguish new files from old files.

Displaying and Setting the Time

To change the time type **TIME** and press ENTER. DOS will display the current time in the format hh:mm:ss.ffp. The hour, hh, is represented by a number from 1 to 12. Minutes are represented by mm and can range from 0 to 59. Seconds are also represented by a number from 00 to 59. The ff entry represents a fractional part of a second. The p is used only when PM is to be appended to the time representation.

If you want to enter a time of 8:30 P.M. you could type **20:30:00** in response to the DOS time prompt. Another option would be typing **8:30:00p** or **8:30p**. Either way you enter it DOS displays your entry as 8:30:00.00p if you redisplay the time.

At a Glance

In this chapter you learned some basics about DOS and how to perform some basic DOS tasks including:

- **Perform a warm boot of DOS—CTRL-ALT-DEL**
- **Stop the current DOS task—CTRL-BREAK or CTRL-C**
- **Activate the DOS Shell at the command prompt—DOSSHELL**
- **Activate the DOS Shell—ALT or F10**
- **Switch to a different area of the DOSSHELL window—TAB or SHIFT-TAB**
- **Exit the Main Program Group—ESC or select Main in program list**
- **Exit to the command prompt and save the DOSSHELL settings—SHIFT-F9**

- **Return to the DOS Shell after a temporary exit—Type EXIT and press ENTER**
- **Exit to the command prompt after closing the DOS Shell—ALT-F4**
- **Select with the Mouse—Click the left mouse button**
- **Determine which release of DOS you are using—VER or use the Shell menu selection Help About Shell**
- **Clear your screen—CLS**
- **Display and change the current date—DATE**
- **Display and change the current time—TIME**
- **Activate a different drive—Type the drive letter followed by a colon**

Chapter 2

Understanding File Basics

Files are the basic unit used to store information on your computer. If you type a memo, create a budget projection, or write a business plan, initially this information will be placed in the memory of your computer. If you decide that you want to maintain a permanent copy of the information, a file must be created on disk. DOS handles the hardware interface to obtain the necessary space for storing this file.

In order to insure easy accessibility to the data at a later time and to provide a consistent way of handling files, there are a few rules that must be followed when naming files for DOS to store on your disk.

What You'll Learn

In this chapter you will learn to:

- **Create valid DOS filenames**
- **Explain some of the types of files that can be created**
- **Create your own file extensions**
- **Establish a set of naming conventions for all of your files**
- **Use wildcards in your filename specifications**
- **Obtain a list of files**
- **Create a text file**
- **Display the contents of a text file**

Understanding What a File Is

When you have a collection of printed information on a topic and want to organize it for easy access, you might put all the different pieces of information into a file folder or notebook. Computer files are similar to folders or notebooks since they contain a collection of related information. This collection might be all the lines you type to create a memo, the assortment of numbers that make up your budget, or information about all the clients you have worked with during the last year. When this information is stored as a file it is written to disk in a consistent fashion and is assigned a name that can be used to access it at a later time. Figure 2-1 shows some different types of information that might be written to a file on disk.

Figure 2-1: Information that Can Be Stored in a File on Your Disk

```
Invitation List

Jim Barker
Carol Deaver
Wanda Evers
Jill Franklin
Paul Mellon
George Nanton
Harry Stone
Virginia Taylor



Sales Projections

            1991        1992        1993        1994        1995
Sales       4,250       5,000       5,500       6,000       8,000
Expenses    3,150       3,500       3,850       4,000       5,000
Profits     1,100       1,500       1,650       2,000       3,000

Dear Mr. Smith:

  We would like to acknowledge your recent order for Haven
crystal.
The baskets that you ordered are currently out of stock.
They will
be shipped to you by early November.

Sincerely,

Carol Lewis
```

Three different types of information: an invitation list, sales projections, and an order receipt memo.

The size of files is measured in bytes. A byte is the amount of space required to store one character such as A, N, 8, or $. The size of files vary greatly depending on the amount of information stored in the file. The size of a file can never exceed the size of the disk on which the file is stored. A file that does not fit on a disk cannot be continued to another disk since it must be completely contained on one disk.

The task DOS has of organizing all the information you enter into files on your computer is similar to the task you face in placing memos or clippings in a file folder. One important difference is that you cannot see the data once it is stored on disk. When you have files stored on disk, you can check with DOS and see that there is information stored under the name of the file, but you cannot easily view this information for many types of files. This inability to work with a physical object adds some of the mystique that often surrounds computer usage for a new computer user. Later in this chapter you will learn a DOS command that allows you to take a look at the contents of at least some types of files that DOS stores for you.

Some files contain nothing but the text characters that you can type from your computer keyboard. Other files are text entries with more information added to define how the text should look when it is printed. You will not be able to view the contents of these files directly through DOS and understand the entries since the additional information is not represented with keyboard characters that you can understand. You will need to use the program that you used when initially creating these files to take this look. Other files are written in a language that only the computer can understand. If you attempted to view the contents of these files you would see nothing but unfamiliar characters on your screen that might look something like this:

Figure 2-2: Viewing a File Written in a Language Only the Computer Understands

```
C:\HSG>type c:\dos\command.com
  \◄                                                  P┐  δ4P┐◙ δ.P┐◙ δ(P┐♥ δ''P┐♦ δ└P┐♣ δ_P┐♠
 δ►P┐ δ
      P δ♦P┐          .ë▲4◙╖♥◙£╥α╥α♥╪¥X£.Ç>3◙ Φ
C:\HSG>
```

Frequently these files are program files that direct your system to execute a series of instructions. You cannot use DOS to obtain any information about the contents of these files. You will learn more about file types in the next section.

■ ***NOTE:*** You will learn to identify many types of files based on the extension added to the filename.

Although files can be stored on a number of different magnetic media, disks have become the most popular method of storage since they support random access to the one file you want to read. Other magnetic media such as tapes require you to start reading at the beginning and continue reading until you obtain all the information you want. This means that you will read many files that are of no interest with this storage method.

To access the file you want from a disk, you will supply the name you assigned to the file when you first created it. To insure that the name uniquely identifies one file, DOS requires you to follow a few rules when you assign names to your files. DOS will consistently retrieve all the information in the file when you specify the filename correctly.

Understanding File Types

There are five basic file types that DOS supports. These are: program files, batch files, data files, formatted ASCII files, and unformatted ASCII files.

Using Program Files

Program files are the files that you execute to have your computer perform a series of instructions. The instructions within the program may complete a simple task such as formatting a disk or they may format the screen as a menu and allow you to select from a range of different options. The program would then handle all the processing for the data that you enter.

Program files other than MS-DOS programs always have a .EXE extension. When you start the WordPerfect program named WP, you are actually starting the program named WP.EXE. Likewise, when you acquire programs for Harvard Graphics, dBASE, and 1-2-3, one of the files on your disk will have an .EXE extension. For the programs listed, the program files are HG.EXE, DBASE.EXE, and 123.EXE.

MS-DOS programs are either internal or external. Internal programs are built into DOS. The capabilities these programs provide are available as soon as you load DOS in your machine. Other programs are external and require DOS to read the program file from disk to carry out the task you requested. External DOS program files have a .COM extension. When you use a DOS command that requires an external file, DOS matches your command entry with the file in your current directory on your DOS disk that consists of the command name followed by .COM.

For example, when you type **FORMAT** at the DOS prompt, DOS executes the program file FORMAT.COM. If you misspell the command name, DOS displays the message **Bad command or file name**.

■ ***NOTE:*** You may want to keep your program files in separate disks/directories from data files to prevent inadvertent damage to the program files, as well as accidental deletion.

In Chapter 5, you will learn how to customize the DOS Shell to add your own programs and program groups to it. If you select a program group from the Shell you will be presented with a list of all the program files within the group that you can execute. When you select a program from the Shell the program's instructions are executed immediately.

Using a .COM or .EXE extension for a non-program file will not allow you to execute the file. A .COM or .EXE file must contain instructions that your computer can understand.

■ ***NOTE:*** .COM and .EXE files are created as the result of a special compilation/assembly process. This process works with source program code written in a programming language such as C, Pascal, or Basic and turns it into code the computer can understand. This code for the computer is stored in a .COM or .EXE file.

Using Batch Files

Batch files contain MS-DOS commands. They are ASCII text files without any special codes added. These files contain nothing except text characters. Each line in a batch file consists of an MS-DOS instruction. The format of these instructions is the same format that you would use at the DOS prompt.

Batch files always have an extension of .BAT. Without this extension DOS will make no attempt to read the file.

Batch files allow you to reuse a series of MS-DOS commands. Instead of typing the commands at the DOS prompt each time you need to perform the same DOS tasks, all you need to do is type the name of the filename from the batch file.

You can store a file named AUTOEXEC.BAT in the main directory of the disk that you use to boot your system. The instructions in this batch file are executed each time you boot DOS.

The instructions in other batch files are only executed when you type the name of the batch file. DOS checks to see if there is a program file that matches the name that you enter and executes the instructions in the program if it finds it.

If there is not a matching program file, DOS checks for a file with a matching name and .BAT as the filename extension. DOS then performs the DOS commands stored within this batch file.

Using Unformatted Text Files

Unformatted text files contain nothing but text characters. They do not have any special formatting codes that programs like word processors add to display text with boldface or underlining.

Many software packages allow you to print reports and other output to an unformatted ASCII file as a means of transferring the data to another program. The data that would normally be printed is written to a file, line for line, the same way that it would appear on a printed page without any of the special formatting or fonts that might be used if the information were displayed directly on the printer.

Batch files are a special case of unformatted text files. They differ from other unformatted ASCII files since they can only contain DOS commands and must have a .BAT filename extension.

■ ***NOTE:*** You can read ASCII text files into many software programs including word processing packages. This allows you to print a report from a spreadsheet or database package and incorporate it within your word processing document.

Using Formatted Text Files

Formatted text files contain text in addition to formatting codes. These formatting codes tell DOS how the text in the file should appear. Word processing programs normally create these formatted text files. If you attempt to use DOS commands to look at the contents of these files, some of the data will be readable and other entries will not be readable. The format codes within the file will appear on your screen as strange characters that are not part of the keyboard character set. WordPerfect, WordStar, DisplayWrite, and Microsoft Word all default to this type of text file. You can override the default and have any of these programs write a text file that is unformatted.

■ ***NOTE:*** To create an unformatted text file with WordPerfect, type your text then use CTRL-F5 Text In/Out followed by 1 for Dos Text and 1 for Save. With WordStar you must request a non-document file before typing text. Use the N option for opening a non-document file. Remember, no matter which word processor that you use, you will need to add a .BAT extension to your filename if you want your unformatted file to be able to be used as a batch file.

Using Application Data Files

Some applications do not use any of the standard file types. These file formats are sometimes documented and, at other times, are maintained as proprietary. Depending on the application, these files may or may not be able to be read by other programs. When the file format is documented it is possible to develop your own programs to work with the data in these files. Database packages, spreadsheets, and accounting packages are examples of types of software that normally each use their own format.

Following File Naming Standards

Every file that you create must be assigned a name. DOS uses this name to identify the file and insure that you access the information you wish to work with. You must follow a few rules when creating filenames to be sure that DOS accepts your entry.

Components of Filenames

The names that you assign your files can have two parts: the filename and a filename extension. Filenames cannot exceed eight characters. Although filename extensions are optional they can consist of as many as three characters. Filename extensions can provide a way to further describe the contents of a file.

If you choose to use a filename extension, it is separated from the filename by a period (.). Once assigned, the filename extension is an integral part of the filename and DOS will not recognize the file unless you supply both parts of the filename. This can sometimes cause problems for you since some of your application programs will automatically add filename extensions. When you attempt to work with these application files from DOS, you will not be able to access them unless you supply both the filename and the extension that the application program added.

Using Allowable Characters

DOS will not accept your filenames unless they conform to a set of pre-established rules. These rules apply to all the filenames that you create. You can use the alphabetic characters A through Z and the numeric digits 0 through 9 in your files names. You can also use the underscore (_), braces ({ }), parentheses (( )), tilde (~), caret (^), ampersand (&), exclamation point (!), pound sign (#), hyphen (-), and dollar sign ($). Other special characters such as | : ; , * ? \ < > " [] / = and + are specifically forbidden since they are reserved DOS symbols used to represent and invoke special DOS features. Spaces and commas should not be used within a

filename. The only period (.) allowed is the one that separates a filename from a filename extension.

- ***NOTE:*** I have found that the best strategy is to avoid the use of all special symbols except the underscore. You will not have to guess which characters are allowed and will not have to worry about compatibility with later releases of DOS.

Each filename in a given location must be unique. Although you can have a file named BUDGET on drive A and another file of the same name on drive B, it is not possible to have two files named BUDGET on a single drive.

Avoiding DOS Names

DOS reserves certain names for its own use. You cannot use any of these words for your own filenames. The list of reserved words includes the following: aux, clock$, com, con, lpt, lst, nul, and prn. These names are devices that you will learn to use for input or output in later chapters. You must realize that you are not working with a file when you send information to these devices.

In addition to DOS reserved words, you should avoid files that have the same names as your programs and batch files. Even though you can use these names with a different filename extension than the one used by the program or batch file, the confusion that can result should be sufficient motivation to avoid this situation.

When you enter a command at the command prompt, DOS searches the current directory for a .COM or .EXE file with the name that you specify. It executes this program if it locates the matching file. If DOS is unable to locate a .COM or .EXE file with the name specified, it looks for a .BAT file with the same name. A .BAT file in the same directory with a .COM or .EXE file with a matching name will never be executed. If you supply a path statement for an extended search, DOS will also search other directories for your files before giving up.

Using Filename Extensions

Filename extensions expand the number of characters you can use to uniquely describe a file from 8 to 11. You can use from one to three characters for each extension. The same character limitations that applied to filenames apply to the extensions.

Conventions for Filename Extensions

There are established conventions used for these filename extensions that place some limitations on their use. DOS always uses .EXE and .COM extensions to represent program files. Batch files always have a .BAT extension. You will want

to avoid the use of these extensions for your data files since their use may lock your system and require you to reboot your computer. Table 2-1 shows some of the common filename extensions and the type of files they are used with.

Table 2-1: Common Filename Extensions

Extension	File Contents
.ASC	ASCII text
.ASM	Assembly language program
.BAK or BK!	Backup file
.BAT	DOS batch file
.C	C language program source code
.CNF	Configuration file to customize a software package
.COM	DOS program file
.DBF	dBASE database file
.DCT	Dictionary file for a word processing package
.DOC	Word processing document file
.EXE	Program file containing executable code
.HLP or .ME	Help file
.MDX	Multiple index file in dBASE IV
.MSG	Message file
.NDX	dBASE index file
.OLD	Backup file
.OVL	Overlay file used by a program
.PCX	Picture file used by PC Paintbrush
.PRN	Print file
.RFT	Revisable form text file used by DisplayWrite
.SYS	System file or device file
.TXT	Text file
.WK1	1-2-3 Release 2.01 or 2.2 file

Extension	File Contents
.WK3	1-2-3 Release 3 file
.WKQ	Quattro Pro file
.WKS	1-2-3 Release 1A file
.$$$	Temporary file

Your Extension	Your Type of File

Accepting Extensions Added by Application Programs

As mentioned earlier, many software programs automatically add a filename extension after the filename that you supply when saving data from the program. For example, if you are working with 1-2-3 and save a worksheet as BUDGET91, 1-2-3 saves the file as BUDGET91.WKS, BUDGET91.WK1, or BUDGET91.WK3 depending on whether you are using 1-2-3 Release 1A, 2.2, or 3.0. When you save a graph with 1-2-3, .PIC is added as the filename extension. dBASE uses special extensions for databases, reports, labels, and the other types of data files that it creates.

Other programs save your files with the exact names you have entered. If you type a file into WordPerfect using the filename MEMO12, no filename extension will be added.

It is important to determine if the software you are using is automatically appending filename extensions to your files. If you attempt to access files from DOS without the filename extension added by the program, DOS will display the message **File not found** since it cannot locate the exact filename that you specified. More than one user has experienced a temporary feeling of panic when the data they had so carefully entered appeared to have vanished from their system. The wildcard capabilities you will learn about later in this chapter will allow you to search for a filename and accept any filename extension thereby helping you to locate these files.

Adding Extensions Yourself

Many of the programs that add filename extensions for you will allow you to use your own extensions if you prefer. Although this seems to restore the flexibility of filename extensions, it often means that the package you are using will not

automatically include the filename when you produce a listing from the program. For example if you save a 1-2-3 worksheet with the name BUDGET91.OLD when you attempt to retrieve 1-2-3 worksheets, this file will not be listed by 1-2-3. You can still access the file, but you must supply the entire filename including the filename extension. Use the empty rows in Table 2-1 to document any filename extensions that you create for your own data.

Defining the Location of Files

Your computer probably has more than one disk drive. You can store data on any of these drives. If you use the hard disk you do not need to insert a disk before accessing the drive since the hard drive is normally not removable.

■ ***NOTE:*** Several years ago, a hard disk drive known as a Bernoulli box was popular. This drive used removable cartridges that had capacities of 5MB to 20MB. If you have this type of drive, you must insure that it contains a cartridge before using the drive since it functions much like a drive used for diskettes.

When you are using a drive that has a removable disk you must place a disk in the drive before attempting to access files on the drive. Failure to place a disk in the drive will result in the DOS error message:

```
Not ready reading drive - Abort, Retry, Fail?.
```

In addition to using different drives, if you have a hard disk you will use directories to organize DOS entries for the various files stored on the disk. Directories are special user-created files that store information about the files managed by the directory. The directory does not contain the data for the files. Directories contain an entry for each filename managed by the directory, the filename extension, the file size, the date and time the file was created, and the physical location on the disk where the data for the file begins.

When you tell DOS to perform a task with a file DOS assumes that the file you wish to work with is in the current drive. The current drive is the highlighted drive at the top of the DOS Shell or the drive indicated by the DOS prompt if you are working from the prompt line. The current directory is the directory on the hard disk whose contents you are currently working with. Although you can use directories to organize a diskette it is seldom done since their capacity is much less.

You can always work with a file that is not on the current drive or in the current directory. If you want to work with a file in a different location you must specify the location of the file when you specify the filename. The location of a file is

called its *path*. This is an appropriate name since it is the path DOS must follow to locate the file.

To specify the path for a file you list the drive letter followed by a colon (D:). This is followed by the directory where the file is located. Later you will learn that directories can form a hierarchical structure with as many as eight levels. For now all you need to remember is that multiple directory levels must be separated from each other by a backslash with another backslash following the last directory entry. To indicate that the file BUDGET.WK1 is in the Lotus subdirectory on drive D you would enter the path as D:\LOTUS BUDGET.WK1. If the same file was in the subdirectory ACCT beneath the LOTUS directory the path would be D:\LOTUS\ACCT\BUDGET.WK1. Table 2-2 provides a look at some additional path specifications for files and the location to which they would direct you.

Table 2-2: More Path Specifications

Path Specification	Location of BUDGET.WK1
A:BUDGET.WK1	Drive A
C:\LOTUS\ACCT\BUDGET.WK1	A subdirectory under the LOTUS directory on drive C
C:\BUDGET.WK1	The main directory on drive C
D:\LOTUS\BUDGET.WK1	The LOTUS directory on drive D
B:\BUDGET.WK1	The main directory on drive B
D:\LOTUS\ACCT\FIS90\BUDGET.WK1	The FIS90 directory located on drive D under the ACCT directory which is under the LOTUS directory

Establishing Your Own Naming Conventions

Although following DOS rules is the first important step to successful file use, you will benefit from adding a few rules of your own. These rules are more informal than the rules established by DOS. Your filename will not be rejected if you decide not to follow these conventions. Since you will benefit from the consistency you will have to monitor your use of conventions.

A consistent set of rules that you can apply when establishing the name for a new file will allow you to read the filename in the future and know what the file contains. It will also allow you to use some techniques covered later in the chapter that allow you to use one command to work with a group of files.

Adopting Simple Rules

The naming conventions that you establish for yourself do not need to be complex. They do require a bit of planning since you will need to make an

analysis of the types of data that you will place in files and establish rules for naming each type of data.

If you need to keep sales records for each of the last three years, you might use SLS89, 90SALES, and SALES91. Although you might eventually guess as to the contents of each file, it requires more effort than a consistent format. A better choice would be to follow a consistent naming approach and use SALES as the first five characters in each of the filenames and add two characters to represent the year numbers. This naming convention provides names of SALES89, SALES90, and SALES91. The consistent pattern tells you at a glance the contents of the files.

If you adopt similar rules for each type of information you store, you will be able to name any file with a few seconds of thought. If you have a variety of information that you must store a sample chart of file naming conventions on the wall next to your computer will help you spot the proper convention for any type of information quickly. Table 2-3 shows what your sample chart might look like.

Table 2-3: Sample File Naming Conventions

Category	Convention	Explanation
Weekly Payroll	PAYMMDDY	Where MM is the month, DD is the date, and Y is the year's last digit
Withholding	WITHMMYY	Where MM is the month and YY is the year's last two digits
Accounts Receivable	ARMMYY	Where MM is the month and YY is the year's last two digits
Accounts Payable	APMMYY	Where MM is the month and YY is the year
Budget	BRRRYY	Where RRR is the three-character region abbreviation
	STH	South
	NTH	North
	EST	East
	WST	West
	YYY	Is the last two digits of the year
Sales	SRRRYY	Where RRR is the three-character region abbreviation

Using a Name and Extension to Determine the Contents of a File

Using the file naming conventions you have established for yourself and the chart of extensions in Table 2-1 you can determine the contents of most files. New programs that you acquire may use extensions you have not encountered in the past. These should be added to Table 2-1 for future reference.

Using Wildcards to Specify a Filename

Anytime you want to work with a file you can specify the filename to insure the selection of the file you want to use. If you need to perform the same task with many files, it can take a significant amount of time to complete your work with all the files. If the files you wish to work with have similar filenames, but they can be distinguished from other filenames, you can use wildcard characters to help you select the entire group of files quickly.

You can think of wildcards like the blank tiles that can be used when playing the word game called Scrabble. In Scrabble the blank tiles can substitute for any letter in the word. The wildcards in DOS are also placeholders that can be filled with any character.

Using an Asterisk as a Wildcard

DOS has two different wildcard characters, the * and ?. The asterisk can be used in place of a filename or filename extension or in place of the characters that end the filename or filename extension. An asterisk can represent one or more characters.

The entry *.* represents any file regardless of its name or extension. The first * will match with any filename of from one to eight characters. The second asterisk will match with any extension of from one to three characters. Files without an extension will also match as long as the filename portion of the entry matches.

You can use an asterisk to replace part of a filename or extension as long as the unknown characters come at the end of the entry. You can use SALES* to indicate that you want all the files that start with SALES. These files can have any entry in the one to three characters range that can follow SALES, SALES90, SALESWST, and SALESN are all filenames that would match the entry.

An asterisk can also replace the end of a filename extension. If you know that the filename is BUDGET and the filename extension starts with a W you might enter BUDGET.W*. BUDGET.WK1, BUDGET.WRS, and BUDGET.WK3 are all filenames that would match your entry.

Using the ? as a Wildcard

A ? can be used in a filename to represent a single character. You can use as many ? entries as you want in any part of a filename. Each ? will replace a single letter in the entry. If you want to work with a file that begins with B and ends in G with two letters between you would use B??G. This entry would match files named BUDG, BADG, and any other filename with two letters between the B and G in the filename.

Looking at the Way Files Are Stored

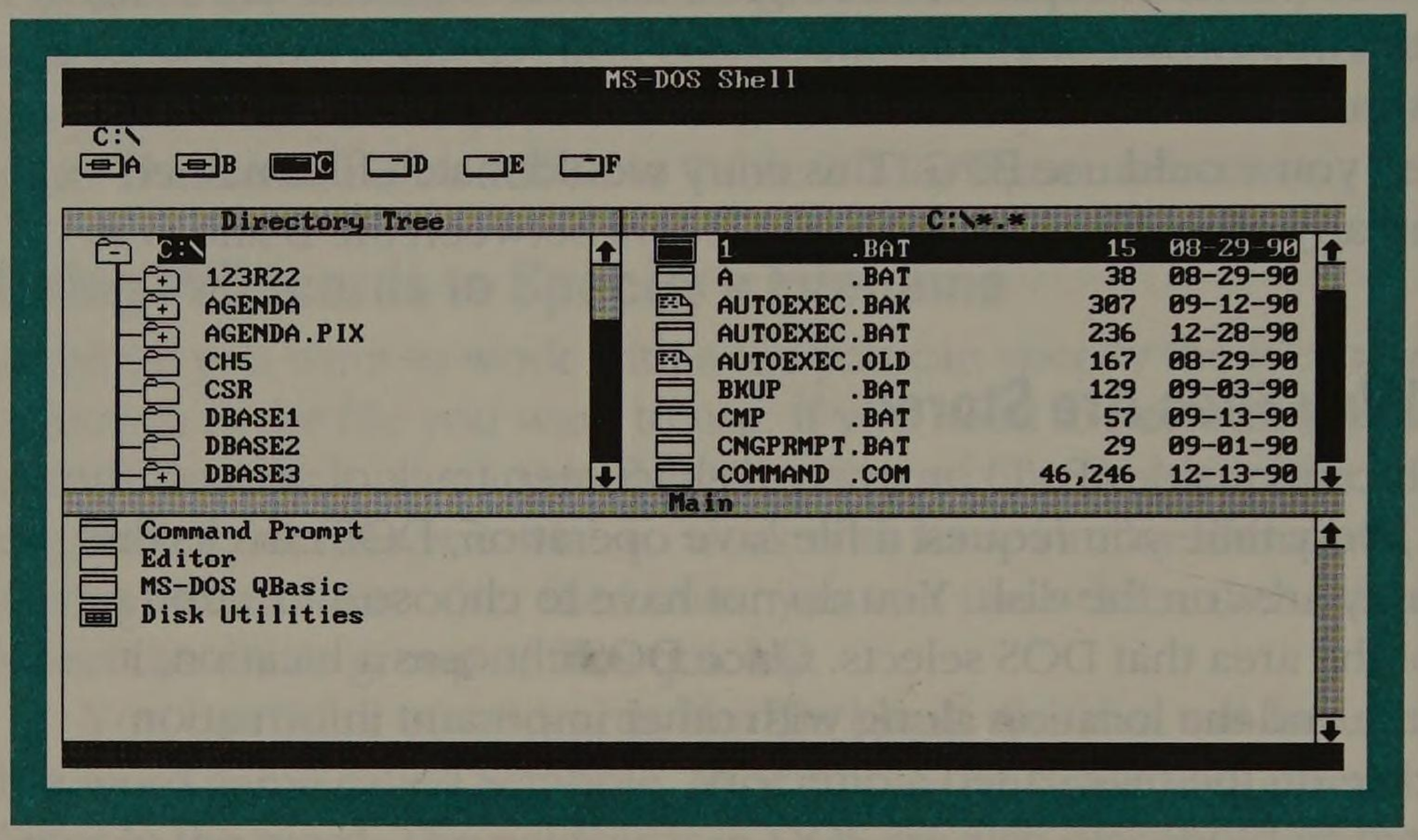
DOS uses a file allocation table (FAT) on every disk to keep track of areas of the disk that are free. Every time you request a file save operation, DOS checks the FAT to find an empty area on the disk. You do not have to choose a location or even keep track of the area that DOS selects. Once DOS chooses a location, it records the filename and the location along with other important information about the file in a file on the disk called a directory.

Understanding How a File is Written to Disk the First Time

When you first write a file to disk it is likely to be quite small, especially if you save periodically rather than wait until you complete an entire project. DOS will probably be able to fit all of your data within the first free area it locates on the disk. Since all the data for the file is stored in one location, access to the data in the file will be quick because DOS can read all the information it needs from that one location.

Looking at the Effect of a Revision on File Storage As you revise a memo or a report, the size of the file needed to store all of your additions may increase. The file may no longer fit within the original area reserved for the file on the disk. As long as there is sufficient additional space on the disk, DOS will check the file allocation table and obtain additional space each time you save the expanded file. Although DOS keeps track of all the additional areas on the disk needed to store the expanded file, it can no longer obtain the entire file in one location on the disk. If there are a substantial number of revisions to many files, the space on your disk can become very fragmented. This will make your system require additional time when you save and retrieve files to the disk. The fragmented pieces can be eliminated by copying a file to another disk then copying the files back to the original disk. You will learn more about this copying technique in Chapter 7.

Figure 2-3: A List of Files in the Shell Window



The shell window is divided into several parts

Obtaining a List of Files

If you are working from the DOS Shell you always have a list of files displayed on the right side of your Shell Window as shown in Figure 2-3. In Chapter 4 you will learn how to make changes to this list to see different file types or files in different locations. If you are working from the command prompt, a file list does not appear. Normally your DOS prompt displays the letter of the active drive but it can also display the active directory on the drive. In this section you will learn how to obtain information on your files while you are working at the command prompt.

Looking at All the Files in the Current Disk/Directory

When you are working at the DOS prompt and want to display all the files in the current drive/directory type **DIR** and press ENTER. DOS lists all the files in a five-column list scrolling information off the screen faster than you can read it when there is a lengthy list of files. The five columns in the directory listing contain the filename, the filename extension, the size of the file in bytes (characters), the creation date for the file, and the creation time. Figure 2-4 shows the last part of a directory listing.

If the entry is for a directory file rather than a regular data or program file, only four entries are made in the columns. Instead of entering a filename extension and a size, <DIR> is entered between columns two and three in the listing.

The end of the directory listing provides additional information. DOS 5 lists the number of files in the current drive directory and the total bytes these files

require for storage. In addition, the number of available bytes on the current disk are displayed.

Figure 2-4: The Last Part of a Directory Listing

```
DF1_3     FST      69440 09-01-90    1:42p
BKUP      BAT        129 09-03-90    9:09p
HELLO     BAT         83 09-03-90    9:50p
REM1      BAT         26 09-03-90    9:44p
PAUSE1    BAT         28 09-03-90    9:45p
ECHO1     BAT         50 09-03-90    9:47p
EXST1     BAK         42 09-03-90   10:00p
SETVAR    BAK         56 09-03-90    9:54p
SETVAR    BAT         53 09-03-90    9:55p
ERRLVL    BAK         76 09-03-90    9:58p
0                      0 09-03-90    9:59p
ERRLVL    BAT         76 09-03-90    9:58p
EXST1     BAT        103 09-03-90   10:02p
CSRTEST       <DIR>      09-03-90   10:02p
PMSDK         <DIR>      09-05-90    1:56p
HSG           <DIR>      09-05-90    1:57p
PIC001    HSG      53338 09-05-90    2:09p
DF1_2     SCR       8128 09-05-90    2:12p
DI2_1     SCR       8128 09-07-90    8:11a
DF2_1     HSG      53254 09-07-90    8:14a
DI2_2     SCR       8128 09-07-90    8:16a
        67 file(s)     472481 bytes
                     13443072 bytes free

C:\>
```

Information in a directory listing is presented in columns. Filename, filename extensions, size, date created, and time created each use a column.

- ***NOTE:*** DOS does not display the number of bytes left in a directory since files are stored on the next available space on the disk regardless of their directory. The only data stored in the directory is the name of the file and other pertinent information for naming and locating the file.

Looking at the Files in Another Location

Although DIR by itself only lists the files in the current directory, you can list the files in any location if you specify the path you are interested in. The path should be separated from DIR by a space.

To look at the files on a diskette when drive C is active you type either:

```
DIR A:
```

or

```
DIR B:
```

depending on which drive you wanted to check.

To look at the contents of another directory on drive C, you can specify the directory after DIR. To look at the LOTUS directory on drive C you would enter DIR C:\LOTUS if drive C was not the active drive and simply DIR \LOTUS if drive C was active.

Viewing a Select Group of Files

You can also look at a selected group of files with the DIR command. Using the wildcard characters discussed earlier in the chapter you can limit the directory display to the files you want to see.

To view a listing of all the .BAT files in the current directory, type **DIR *.BAT**. The * indicates that you want to list the files regardless of their filename and as long as they have a .BAT extension.

To look at all the filenames that begin with PR, type **DIR PR*.***. The first * indicates that any ending for the filename is acceptable and the second asterisk indicates that any extension is acceptable.

The ? character is also supported. To look for filenames where you specify all the positions in the name except the fourth position you might have an entry like **DIR REG?SALE.WK1**. This entry would match with REG1SALE.WK1, REG2SALE.WK1, and similar filenames. You can combine both the ? and the * with a directory request as in **DIR REG?SAL.W*** when you want to be less specific about acceptable filename extensions.

Each of the options you just looked at for the DIR command are different ways of specifying a parameter for DIR. In Chapter 1 you learned that some commands support parameters after the command name to allow you to define the object or group of objects you want the command to work with.

Using All the Directory Options

The DIR command has a number of different options that allow you to create directories in different ways. In Chapter 1 you learned that these options are specified with switches consisting of a slash followed by a letter or number. The directory command supports seven different parameters. You can use these parameters individually or in combination.

Limiting the Display If you request a lengthy directory listing without a parameter, the listing scrolls off your screen before you are able to read it. You can add /p after your directory request as in **DIR B*.* /p** to have DOS pause after displaying the first screen of information. The page size for this screen display will be 24 lines unless you redefine the page size with ANSI.SYS, which is covered in Chapter 15. To display subsequent pages, you can press any key.

Using a Wide Display If you are not interested in file sizes or creation dates or times, you can maximize the number of filenames displayed on a line with the /w switch. Using this option will cause DOS to display as many as five filenames and extensions on a line rather than the single filename per line format without the use of this switch. Directory files are shown in brackets in a wide directory list. Your listing might look something like Figure 2-5.

Figure 2-5: Your Directory Listing

```
C:\>dir /w

 Volume in drive C is DOS400
 Volume Serial Number is 5C38-3850
 Directory of C:\

COMMAND.COM     D1_1.PIX        CONFIG.SYS      [DOS]           [RUP1]
[INSET]         [123R3]         COPYMEMO.BAT    [WS5]           THIS.BAT
[CAPTURE]       A.BAT           [MICE]          [WP51]          MENU.BAT
[XDOS.1]        Z.BAT           COPYMEMO.TXT    LIST.BAT        AUTOEXEC.BAT
SCREEN00.PCX    DIDLY.BAT       AUTOEXEC.OLD    TEST1.TXT       TEST2.TXT
TEST3.TXT       1.BAT           F.BAT           CONFIG.OLD      TEST4.TXT
COPYMEM2.BAT    Z               LK              CNGPRMPT.BAT    ENVIRON.BAT
PRINTALL.BAT    PRNTALL2.BAT    DOFILE          DOFILE.BAK      DOFILE.BAT
[HOT]           DF1_1.PIX       DF1_2.PIX       DF1_3.PIX       DF1_1.FST
DF1_2.FST       DF1_3.FST       BKUP.BAT        HELLO.BAT       REM1.BAT
PAUSE1.BAT      ECHO1.BAT       EXST1.BAK       SETVAR.BAK      SETVAR.BAT
ERRLVL.BAK      0               ERRLVL.BAT      EXST1.BAT       [CSRTEST]
[PMSDK]         [HSG]           PIC001.HSG      DF1_2.SCR       DI2_1.SCR
DF2_1.HSG
       66 file(s)     464353 bytes
                    13451264 bytes free

C:\>
```

With a wide directory listing only filenames and filename extensions are shown, although many more files fit on the screen.

Controlling the Display Based on File Attributes Files have attributes assigned that allow you to categorize them. Some files are classified by DOS as system files which are only for use by DOS. Other files are flagged as hidden files with a special attribute setting. Hidden files and system files normally do not display in directory listings. Files can also be flagged as ready for backup or archive, directory files, and read-only files with an attribute setting.

You can check the attribute setting for files and allow it to control the directory display. To specify that you want to check the attributes for files you will use the /a or /a: switch with one or more of the attributes in Table 2-4. Spaces are not used to separate these options even if you specify multiple attributes. To indicate that you want to see all read-only files that are ready for archiving you would type **DIR /a:ra**.

Table 2-4: Attribute Specifications to Control Directory Listings

Attribute	File Selection for the Directory Listing
/a	All files including system files and hidden files
/a:h	Hidden files
/a:-h	All files that are not hidden
/a:s	System files
/a:-s	Non-system files
/a:d	Directory files
/a:-d	Files that are not directory files
/a:a	Files marked for archive
/a:-a	Files that have already been archived
/a:r	Read-only files
/a:-r	Files that are not read-only

Controlling the Sort Order Directory listings display in the order in which files were added to the directory. If you want to change the display order you can use the /o or /o: switch with the sort order options shown in Table 2-5. Entering **DIR /o:-n** will produce a reverse alphabetical list.

Table 2-5: Options for Specifying the Sort Order

Sort Order	Effect on the Directory Listing
/o	Directories listed before files, alphabetically sorted by filename
/o:n	Alphabetical by filename
/o:-n	Reverse alphabetical by filename
/o:e	Alphabetical by extension
/o:-e	Reverse alphabetical by extension
/o:d	Date and time with the earliest date first
/o:-d	Date and time with the latest date and time first
/o:s	Size with the smallest first
/o:-s	Size with largest first
/o:g	Directories before files
/o:-g	Directories after files

Using Other Customizing Options You can indicate that you want a directory listing of the specified filenames in the current directory and all lower directories if you use the /s switch.

You can display an abbreviated directory listing that consists of filenames and extensions, yet continues to display the entries one per line rather than the wide format if you use /b in your request. The final switch option is requesting the directory listing in lowercase. This is requested with the /l option.

You can combine these options with earlier options in a request that might look like this:

```
DIR /a:r /o:n /b /l
```

which requests an abbreviated lowercase listing of read-only files in alphabetical order.

- ***NOTE:*** Although you can use a number of switches with a directory without leaving a space before each new switch, you will find that the space makes the entries more readable.

Reversing Default Directory Settings In Chapter 15 you will learn that one of the customizing changes you can make to DOS is presetting options for all directory listings. The same switch options available for an individual directory listing can be set as a default for all directory listings. If you want to reverse one of the default settings for an individual directory listing you would use the same switch bit, but precede the letter defining the option with a minus sign. For example, if /p is set for the default and you wish to eliminate this setting for the current listing you would type **DIR /-p**.

Creating a Text File

There are a number of different ways to create a text file. You can use a word processing package as long as you tell the package to save your file without the special codes that are normally added automatically to control the appearance of your document. In Chapter 11 you will learn how to access the DOS System Editor from the Shell to enter files in a full screen editor similar to your word processor. You can also use the COPY command to have DOS copy from your keyboard to a file.

Learning the Steps for Using COPY CON

To create a short, quick, text file, the easiest approach is to take advantage of the device name for the keyboard and copy what you type from the keyboard into a file. You need to type **COPY CON TRY.TXT** and press ENTER to copy what you type at the keyboard into a file on the current drive and directory named TRY.TXT. The DOS prompt disappears from your screen. DOS is waiting for you to enter data from your screen.

After you type each line in the new file, press ENTER to have DOS check the line for syntax errors. After processing your last line, press F6 and DOS will generate a ^Z on your screen. When you press ENTER, DOS will tell you that it copied one file for you.

Trying an Example

Entering a text file requires only a few minutes if it is short. Follow these steps to create a sample:

1. Type **COPY CON NONSENSE**.
2. Type **MY NAME IS**, press the SPACEBAR, then type your name before pressing ENTER.
3. Type **My favorite food is chocolate** and press ENTER.
4. Press F6 to generate ^Z then press ENTER.

DOS will display the message **1 file copied**. You can check for the existence of this file by typing **DIR N*** and pressing ENTER.

Looking at the Contents of Text Files

You can depend on filename extensions to tell you which files contain text and which ones contain programs or specialized application software. Until you have worked with filename extensions a little more you will make some mistakes but attempting to look at a file that does not contain text will not damage the file. The command that allows you to look at the contents of a text file is TYPE.

To look at the contents of NONSENSE on your screen, type **TYPE NONSENSE** and press ENTER. DOS will display your entries back to you with a display that will looks something like the display shown in Figure 2-6 on the following page.

Figure 2-6: The Contents of NONSENSE

```
C:\DOS>COPY CON NONSENSE
MY NAME IS JOE SMITH
My favorite food is chocolate
^Z
        1 file(s) copied

C:\DOS>TYPE NONSENSE
MY NAME IS JOE SMITH
My favorite food is chocolate

C:\DOS>
```

DOS entries showing the creation and display of the file NONSENSE.

At a Glance

In this chapter you have learned the rules you need to follow for naming files. Regardless of the package you create the file with, you must follow these rules:

- **Filenames must not be longer than eight characters**
- **A-Z, 0-9, and _ are the preferred characters for use in filenames**
- **Spaces, DOS reserved words, and a number of symbols are not allowed**
- **An optional one to three character filename extension can be added if separated from the filename by a period (.)**

You have also learned the following facts:

- **Use an * in a filename to represent a group of characters at the end of the name or extension**
- **Use a ? in a filename to represent a wildcard single character**
- **List files on the current drive/directory—DIR**
- **Create a text file—COPY CON filename**
- **Display a text file—TYPE filename**

Chapter 3

Understanding Disk Basics

It is hard to believe that not many years ago, computers used an 80-column punch card for computer readable information. Although not as thick as a disk, the surface area of these cards was larger than many of today's disk drives, yet they only held 80 characters. Today's disks come in many different sizes, but even the smallest capacity disks will hold over 360,000 characters.

Although some new storage media such as CD-ROM may be the key to increased storage capacity in the future, today's computing environment relies on disks for information storage. The low-cost, reliability, and reusable nature of disks puts them in first place as a permanent storage medium. You will want to understand some of the basic terminology and basic techniques for working with disks since they are likely to be your sole source of permanent storage.

What You'll Learn

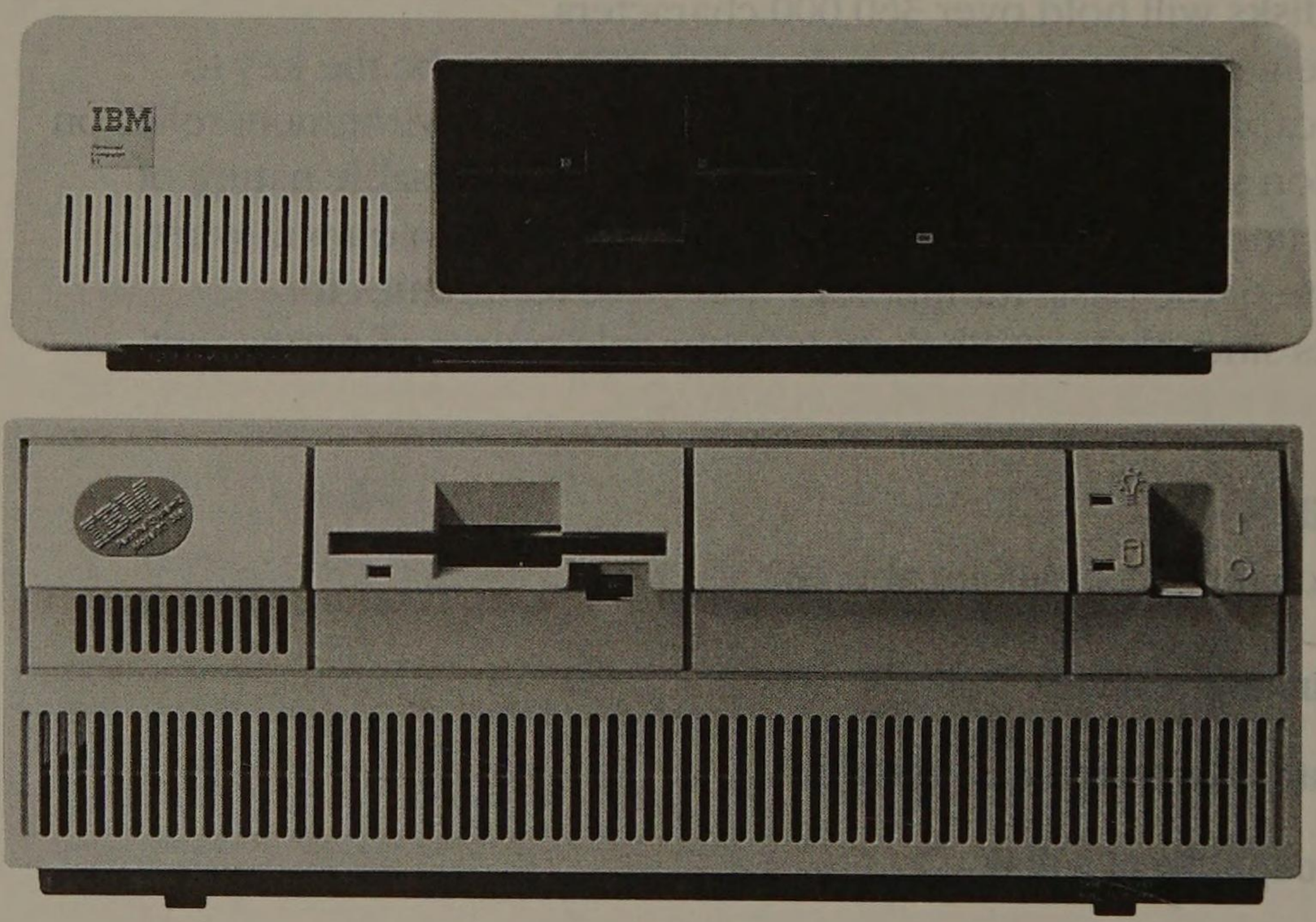
After reading this chapter you will be able to:

- **Explain the structure of a disk**
- **Protect your data from accidental deletion**
- **Determine the capacity of a disk**
- **Explain options for organizing disk storage**
- **Make a copy of a disk**
- **Compare two disks to determine if they are identical**

Typical Disk Configurations

Most computer systems sold today have at least two disk drives. One is a fixed disk drive built into the system unit. There is also at least one removable disk drive. The removable disk drives are called removable because the disks you use with them can be removed from the drive. These drives are also commonly called floppy disk drives. Figure 3-1 shows an IBM XT's 5 1/4 inch floppy disk drive and an IBM PS/2 3 1/2 inch drive. Figure 3-2 shows a system configuration of a 5 1/4 inch and a 3 1/2 inch drive, and an external 3 1/2 inch drive.

Figure 3-1: Popular System Configuration



IBM XT 5.25 floppy drive and PS/2 3 1/2 inch floppy drive.

Figure 3-2: External Drive Attached to the System Unit

A cable connects the external drive to a port on the back of the system.

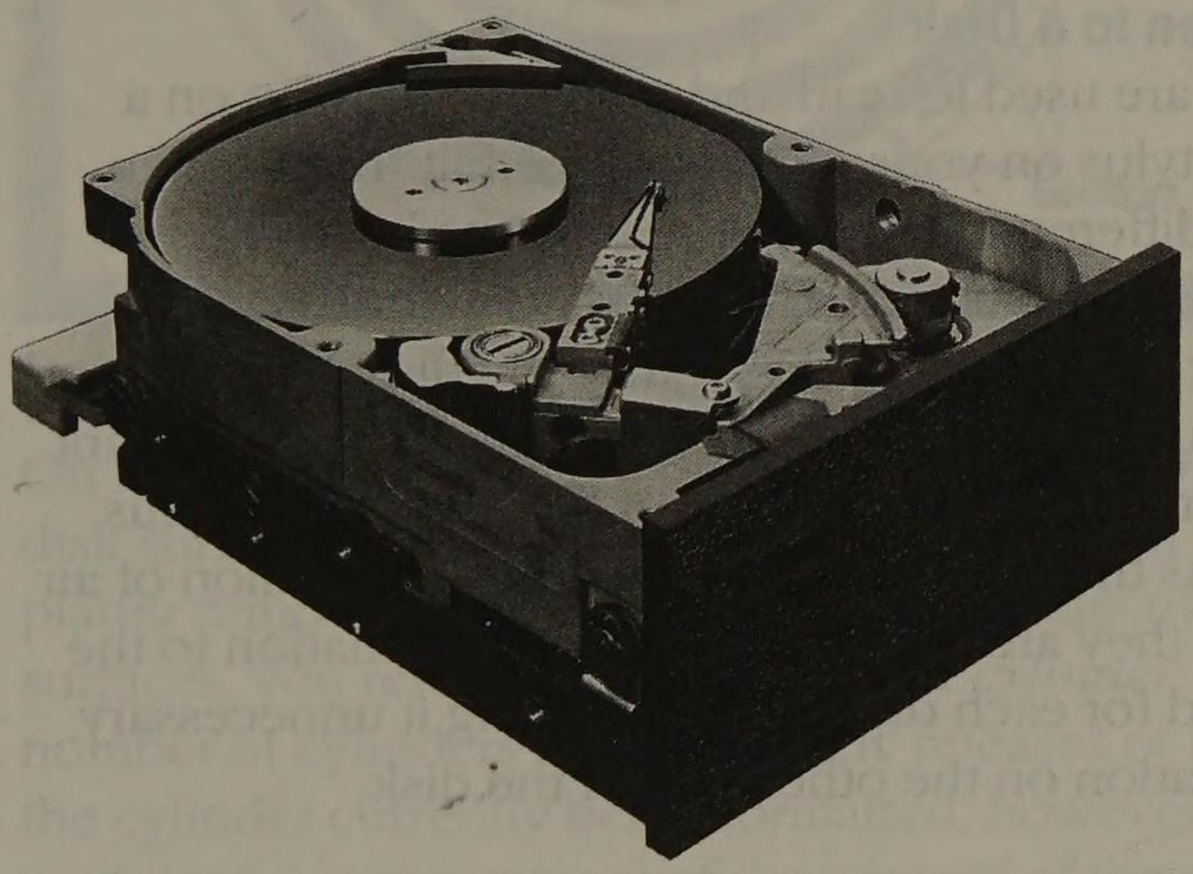
On many systems, the high capacity fixed disk holds in excess of 40 million characters. Removable disks are either 5 1/4 or 3 1/2 inches square with capacities from 360K to 1.44 MB. No matter which type you are working with you have a substantial storage capacity for both program files and data files on your system.

Examining Disk Structure

You cannot see the surface of your hard disk since it is inside your system unit. Even with the exposed oval cutout area on a 5 1/4 floppy disk, you can see only a small part of the disk surface. In both types of disks, the disk itself is a circular platter with a magnetic surface.

A fixed disk drive normally contains several platters stacked one on top of the other with only a small separation between them. If you were to see the fixed disk inside of your system, it might look something like Figure 3-3. Read/write heads on arms can be moved to access any area of the disk. See the section Understanding How Data Is Written to a Disk for more information on accessing disks.

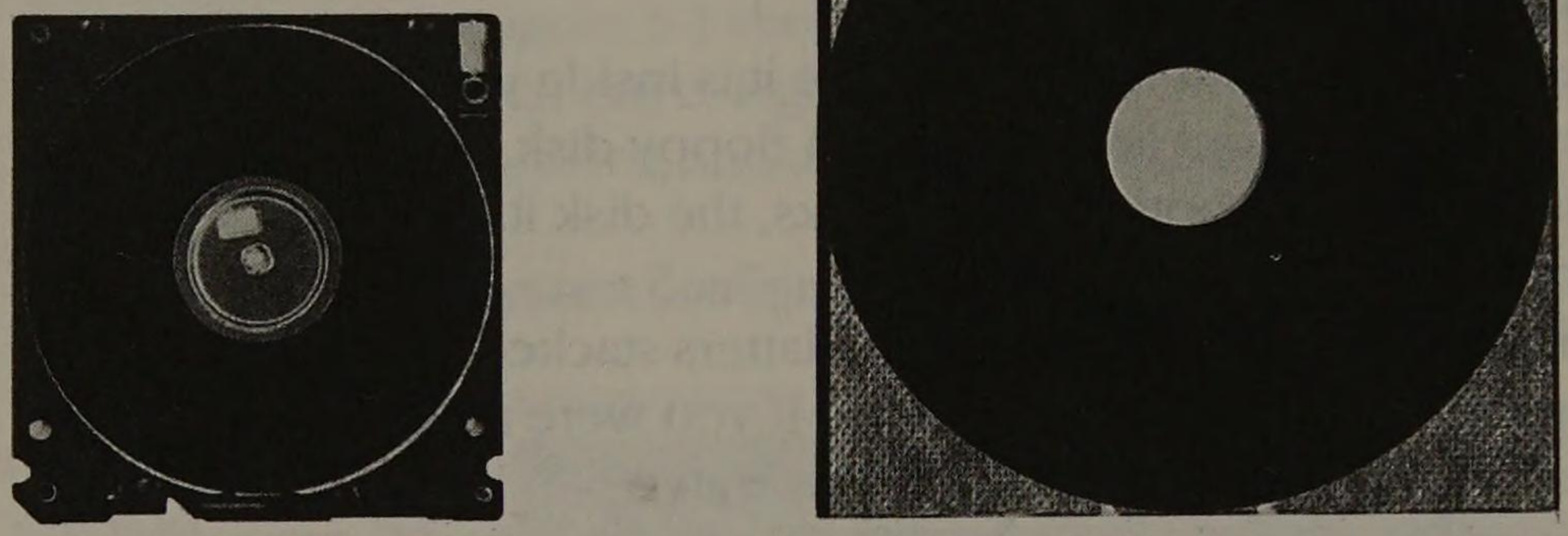
Figure 3-3: Fixed Disk from Inside the System's Unit



A fixed disk is compact despite its storage capacity.

The removable disks have a protective covering that is square in shape. This covering insures that the disk slides into the drive properly. The disk beneath the covering looks much like the platters in a fixed disk except the material is not as rigid. Figure 3-4 shows both disk types. A center hub on the 5 1/4 inch disk is grasped by the drive to allow the disk to spin inside its protective covering. On the 3 1/2 inch disk the metal circle visible on back of the disk has a hole for the disk spindle which allows it to spin within its cover.

Figure 3-4: 3 1/2 and 5 1/4 Inch Disks



Differences in size and type of 3 1/2 versus 5 1/4 inch disks.

■ ***NOTE:*** Fixed disks are often called hard disks because of the hard material in the disk platters. Removable disks have been called floppy disks because the disk itself is very flexible and flops around if removed from its protective covering.

Understanding How Data Is Written to a Disk

Read/write heads attached to arms are used to read and write information on a disk. The concept is similar to the stylus on your record player that "reads" the music from the record. One major difference is that the read/write head can be positioned to read or write data in any location on the disk—without having to proceed from the outer to the inner circles on the disk. Since the disk spins around at hundreds of revolutions per minute, the delay to reach the exact data you want is insignificant. Another major difference is that a needle at the end of the stylus actually rests on the record, whereas the read/write heads float on a cushion of air above the surface of the disk when they are reading or writing information to the disk. There is also a read/write head for each disk surface making it unnecessary to turn a disk over to access information on the other side of the disk.

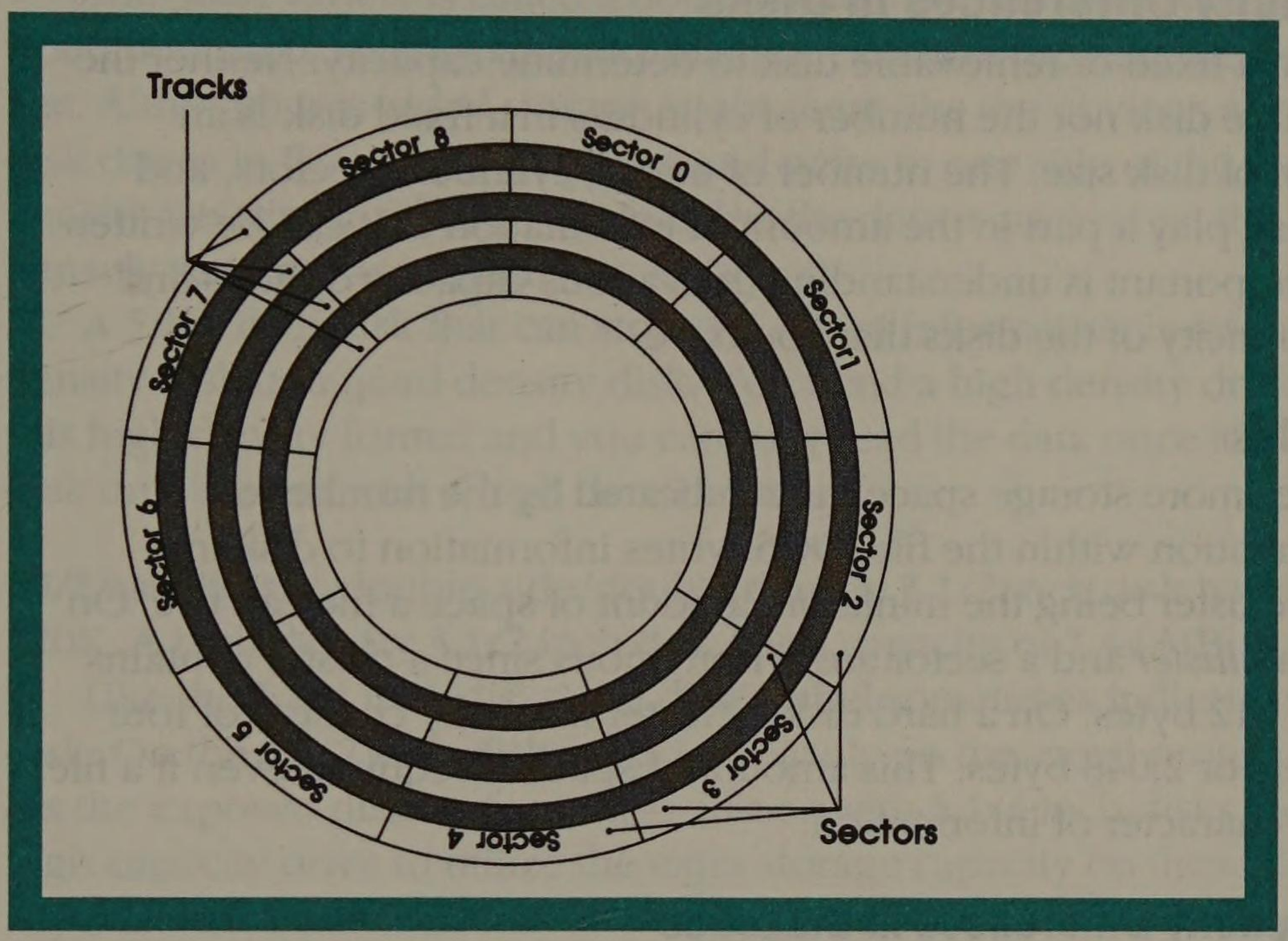
Understanding What a Track Is

Tracks are concentric circles around a disk that indicate where data can be stored on the disk. Figure 3-5 shows the way the concentric circles are placed around the disk surface. If you have been to a theater in the round you can think of tracks that ring the disk hub or center as similar to the rows of seats that ring the stage.

Tracks are not on a disk when you first purchase it. Tracks must be placed on a disk with the operating system. Different operating systems place a different number of tracks on a disk of the same size, partially explaining why disks from different operating systems are not interchangeable.

DOS 5 places a format pattern on 40 tracks of a double-sided/double density 5 1/4 inch disk and 80 tracks of a high density 5 1/4 inch and both types of 3 1/2 inch disks.

Figure 3-5: Tracks Around the Disk Surface



Tracks are concentric circles around the disk.

Understanding What a Cylinder Is

On all disks except the early single sided type, tracks are placed on more than one disk surface. At a minimum, there are tracks on the top and bottom of a single platter removable disk. When you look collectively at the same track on all disk surfaces, you refer to these tracks as a *cylinder*. This means a disk has the same number of cylinders as tracks. Early releases of DOS used to keep you informed of the cylinder currently being formatted, however DOS 5 now provides the more meaningful measurement of the percent of the disk that has been formatted.

Understanding What a Sector Is

Sectors are pie-shaped wedges that divide a disk into sections of equal size as shown in Figure 3-5. If you look at the seats in the theater example mentioned earlier, sectors are like the seat sections that divide the theater into sections A, B, and C. Theater seats in sports coliseums and theaters are often color coded in the different sections.

Like tracks, sectors are not on the disks when you buy them. The formatting process adds a pattern to the disk that also marks these sectors. There may be 9,

15, or 18 sectors per track with 512 bytes in each sector on a floppy disk. A hard disk may vary in the number of sectors they have. Two different models I recently checked had 17 and 32 sectors. Your hard disk is likely to have more bytes per sector with 512 bytes a common size.

Exploring Capacity Differences in Disks

You cannot look at a fixed or removable disk to determine capacity. Neither the size of the removable disk nor the number of cylinders in a fixed disk is an accurate indication of disk size. The number of tracks, cylinders, sectors, and recording density all play a part in the amount of information that can be written to a disk. What is important is understanding the various capacity options and determining the capacity of the disks that you have.

Space Used by Files

A file often requires more storage space than indicated by the number of characters of information within the file. DOS writes information to disk in clusters, with one cluster being the minimum amount of space a file can use. On removable disks, a *cluster* and a sector are synonymous since a cluster contains only one sector of 512 bytes. On a hard disk a cluster normally consists of four consecutive clusters or 2,048 bytes. This amount of space is required even if a file contains only one character of information.

Looking at the Capacity Differences in Diskettes

The two most popular sizes in diskettes are 5 1/4 inches and 3 1/2 inches. Even though these two sizes dominate the removable disk market they are not all alike and vary in terms of their capacities. Table 3-1 provides a summary of the capacity of various disks.

Table 3-1: Disk Capacities

Size	Sides Used	Tracks	Sectors	Capacity	Example of a System
5 1/4	1	40	8	160K	IBM PC
5 1/4	2	40	8	320K	IBM PC/XT
5 1/4	2	40	93	60K	IBM PC/XTDOS 2 +
5 1/4	2	80	15	1.2 MB	IBM AT
3 1/2	2	80	9	720K	PS/2 Model 25
3 1/2	2	80	18	1.44 MB	PS/2 Model 50

5 1/4 Inch Disks A 5 1/4 inch disk used on most computer systems today can hold between 360K and 1.2 MB. Although the disks will sometimes have labels to indicate their capacity, the only sure bet is the box that the disks came in. The higher capacity disks can only be read or written to on a disk drive that supports this higher capacity.

The 360K variety is called a double-sided/double density disk. The double-sided descriptor refers to the fact that both sides of the disk are used for data storage. Although two-sided storage might seem like the obvious approach, the first disk drives in PC's could only read and write to one side of the disk. The double density descriptor refers to the fact that the data is packed on the disk twice as densely as an earlier design.

A 5 1/4 inch disk that can store 1.2 MB of information is referred to as a high density disk or a quad density disk. You need a high density drive to write data in this high density format and you can only read the data once it is written to the disk on a system with a high density drive.

3 1/2 Inch Disks A double-side/double density 3 1/2 inch disk has a capacity of 720K. A high density 3 1/2 inch disk has a capacity of 1.44 MB.

Like the 5 1/4 inch disks, the disk's label sometimes indicates capacity on the disk. On the 3 1/2 inch disks, this is typically on the metal protective slide that covers the exposed disk surface. Also, just as with 5 1/4 inch disks, you must have a high capacity drive to utilize the extra storage capacity on these disks. High density 3 1/2 inch drives normally have the number 1.44 embossed on the drive's disk eject button.

Watching the Evolution of Large Capacity Hard Disks

Several years ago the most popular capacity for hard disks was 10 or 20 MB. Today most new systems have a fixed disk with a capacity of 40 or 60 MB. Hard disk drives that can store 120 MB or even 250 MB are available, but currently their use is restricted to networks where these large capacity disks function as file servers to fulfill the storage needs of a number of network users that all share the same disk space.

As the manufacturing costs of the larger capacity disks decline you will begin to see more of the larger capacity disks in stand-alone PC's. As software has become more sophisticated, the need for additional program storage space has increased dramatically. Word processing packages are now bundled with a spelling checker and thesaurus and database programs have sophisticated reporting programs. All of these additional features require the storage of additional program code to implement their capabilities.

Determining the Capacity of a Disk

You can use the Shell Display Information command or CHKDSK at the DOS prompt to determine the capacity of both fixed and removable disks that have been prepared for storage. In effect, the CHKDSK command can answer all your capacity questions—except for determining whether you have a double sided/double density or a high density drive for removable disks. You will need to try reading a high density disk or format a blank disk in order to determine the type of drive that you have. Instructions for formatting a disk are in Chapter 6.

- ***NOTE:*** Your hard disk may be larger than you think if it was originally formatted with DOS 3.3 or lower. Since these releases only allowed a 32 MB partition, a larger hard drive had to be partitioned. This means you may have 30 MB on drive C and another 30 MB of storage space on drive D.

Working from the Shell to Check Disk Capacity You can check the capacity of a disk with the Shell if you first make the drive the current drive. Use your mouse to click the drive designator boxes or activate the drive area in the DOS Shell window and use the ARROW keys to highlight the drive you want to check before pressing ENTER. Once the drive is active, click Options Show Information or press ALT-o and type an **s**.

Working from the DOS Prompt to Check The Capacity To use CHKDSK all you need to do is enter CHKDSK to check the current drive. DOS displays information on the capacity of the disk as well as the number of files of various types and the number of directories on the disk. Figure 3-6 provides a look at the output of this command.

Figure 3-6: Output from CHKDSK

```
C:\>CHKDSK

Volume DOS400      created 08-03-1990 6:30p
Volume Serial Number is 5C38-3850

  33462272 bytes total disk space
     73728 bytes in 2 hidden files
     59392 bytes in 20 directories
  22040576 bytes in 910 user files
     16384 bytes in bad sectors
  11272192 bytes available on disk

      2048 bytes in each allocation unit
     16339 total allocation units on disk
      5504 available allocation units on disk

    655360 total bytes memory
    577696 bytes free

C:\>
```

CHKDSK provides information about memory as well as the disk.

To check the capacity of a drive other than the current drive, enter CHKDSK followed by the drive specification. An entry of CHKDSK B: will display information on the disk in drive B.

Exploring Ways to Organize Information

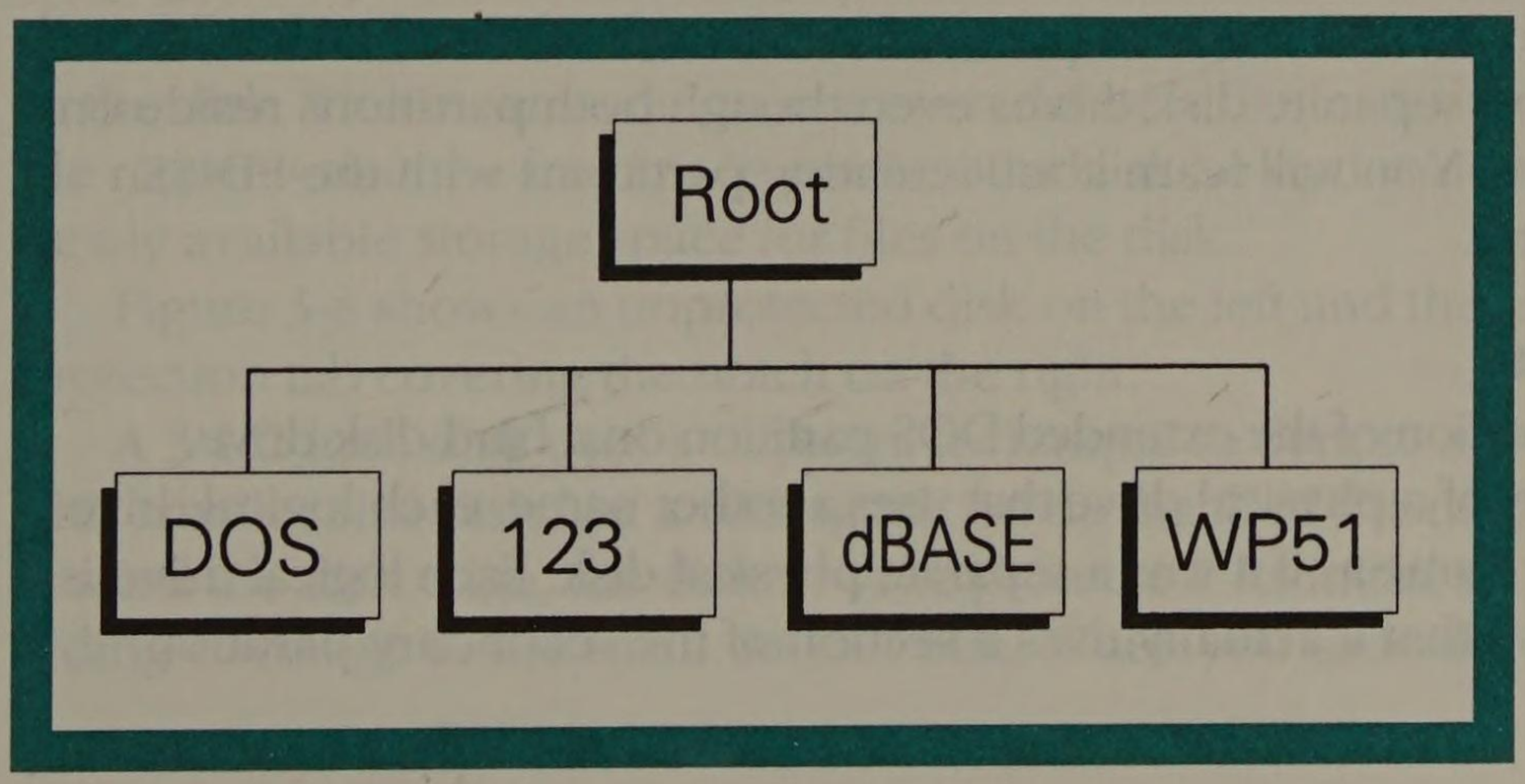
One of the things that DOS can help you with is organizing the vast data resources that you are likely to acquire with continued use of your computer system. You will want to use the most effective organization possible since it will enable you to find files more quickly and will allow your programs to operate without performance problems. Although the details of some of these techniques will be reserved for later chapters, it is important to understand many of them as you explore the capabilities available to you on today's high performance, high capacity disks.

Using Directories

Directories are logical divisions of a disk. Creating directories uses a small amount of space on your disk but it does not physically segment the disk into sections. You use directories to organize a disk since they provide the information on how to locate a specific file on the disk.

Directories logically arrange the files on the disk into as many as eight hierarchical levels depending on the structure you need to organize your information. Directories keep the file information presented at any one time to a manageable level and allow you to access the correct data quickly. One option for creating a logical disk structure is to place application software-oriented directories beneath the root level directory as shown in Figure 3-7. You will learn all the specifics of directories in Chapter 8.

Figure 3-7: Several Subdirectories Beneath the Root Directory



Subdirectoris shown are all on the same level beneath the root directory.

Realizing the Limitation of Root Directory Storage Subdirectories are also essential since the root directory of a disk may not allow you to utilize all the space on a disk. The root directory has limits on the number of files it can manage and it does not allow you to add files once this limit is reached. This limit is dependent on the size of the disk. This means you could have an adequate amount of storage space left on a disk yet not be able to create another file on it. The message that DOS will display when you try to create a file on a disk where the directory is full is:

```
File creation error
```

You will not be able to create another file on this disk unless you first delete a file. The restriction also applies to the creation of a subdirectory since subdirectories require the creation of a file in the root directory of a disk.

If you are using a 5 1/4 double sided disk, you cannot store any more than 112 files in the root directory. A 3 1/2 inch disk supports only 224 files, and a hard disk will accept 512 file entries. These restrictions do not apply to subdirectories on these disks.

Using Partitions

Partitions are physical divisions of a hard disk to divide it into separate sections. Before DOS 4.0, you had to establish partitions to utilize all the space on a hard disk that exceeded 32 MB in capacity. This was because 32 MB was the largest size partition supported. Both DOS 4 and 5 support a partition size that includes all the space on the disk, making it unnecessary to create separate partitions to fully utilize a disk. You will learn how to create partitions on a disk in Chapter 6.

Partitions are still useful when you want to use more than one operating system and your selections do not use compatible file storage formats. You can partition the disk into a primary partition and one or more secondary partitions. You can utilize the primary partition for DOS and other partitions for an operating system such as UNIX or as extended DOS partitions. DOS will think of each of the partitions as if they are separate disk drives even though both partitions reside on the one physical drive. You will learn about creating partitions with the FDISK command in Chapter 6.

Using Logical Drives

Logical drives are a portion of the extended DOS partition on a hard disk drive. Although they are part of a physical drive that uses another name, each logical drive is assigned a different name as if it was a separate physical disk. Each logical drive is also assigned a size so that it actually uses a section of the secondary partition.

If you partition your hard disk, you might use part of the secondary partition as drive D and another part as drive E. If you want to assign additional logical drives you will need to alter CONFIG.SYS to allow a higher letter drive to be assigned than the E since E is the last drive allowed with the default settings. The maximum number of logical drives is 23 if you use the command LASTDRIVE=Z in your CONFIG.SYS file.

The FDISK program covered in Chapter 6 is used to create logical drives. Once created you use the drive letter designations just as you do for drives A, B, or C. You can even create subdirectories on these logical drives as you probably will do with drive C to provide better organization.

Protecting the Contents of a Disk

After you have worked with your computer for a while you will begin to realize that you have a tremendous investment in the data you have stored on your disks. It would be impossible to reenter all of your data in a reasonable time and you have probably already taken precautions to insure you are backing up vital files. If you have not created backup copies of your data, you will want to be certain to follow the directions in Chapter 7 so that you have an extra copy of your entire disk. Although these backups will be invaluable in the event of data loss, you will still want to take precautions to make sure that the loss never occurs since backups are never completely up-to-date. You will want to protect an entire disk if there is no need to change it and where you cannot protect the entire disk you will want to insure that vital files are not accidentally deleted or changed.

Protecting an Entire Floppy Disk

Both 5 1/4 disks and 3 1/2 disks allow you to protect all the files they contain. This process is called *write-protecting the disk*.

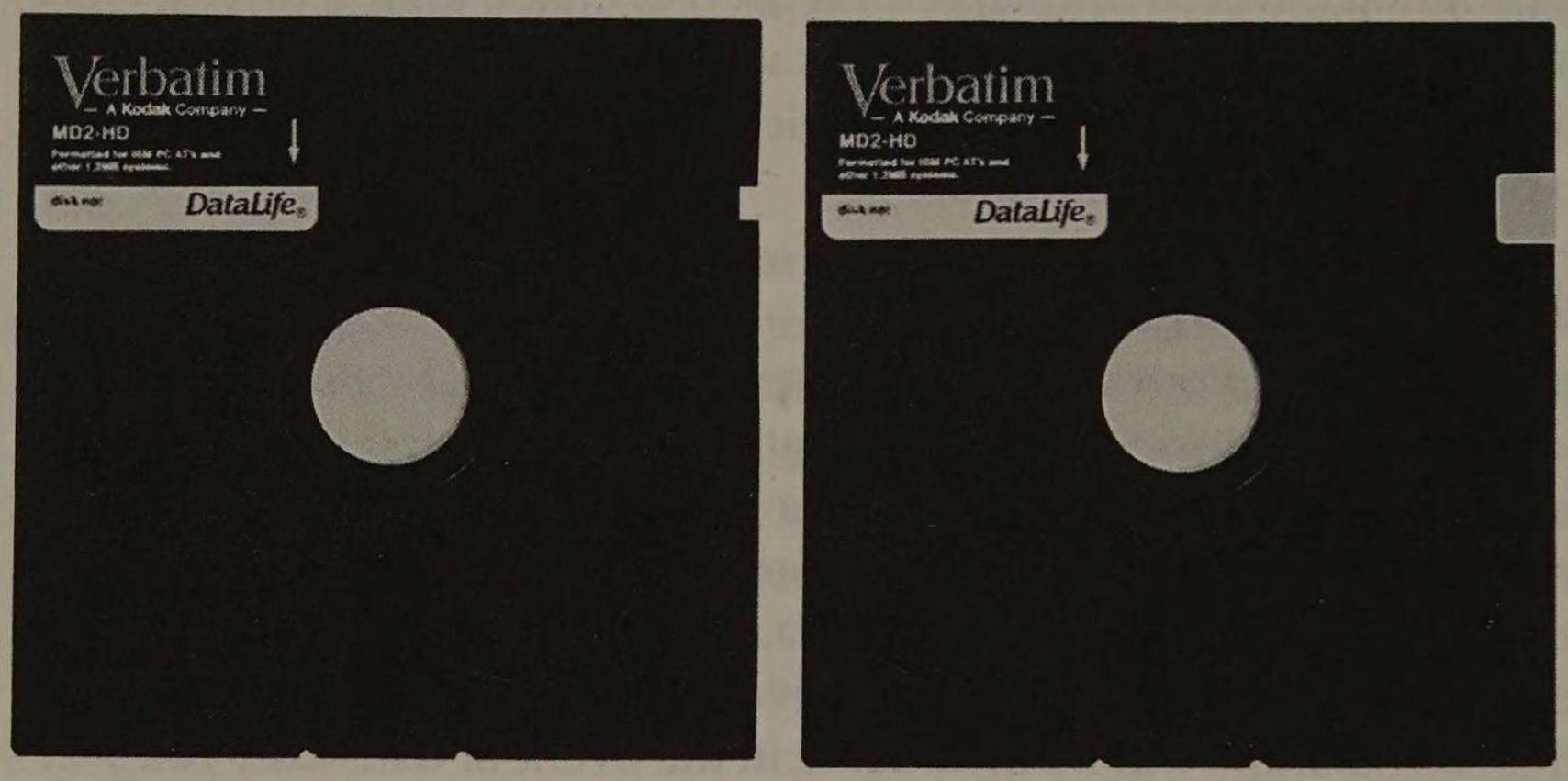
To write protect a 5 1/4 inch disk, you wrap one of the adhesive tabs over the notch on the side of the disk. When this notched area of the disk is covered, the disk drive can sense this "protected" condition and will not write any information to the disk. This means you cannot even delete a file from the disk since deleting a file requires that the free space pool on the disk be updated to note the location of newly available storage space for files on the disk.

Figure 3-8 shows an unprotected disk on the left and the same disk with the protection tab covering the notch on the right.

A 3 1/2 inch disk uses a different technique to denote a write-protected state. This smaller size disk has a sliding tab on the back of the disk. If you slide it toward the disk edge, the disk is write protected. Its initial state is at the top providing coverage to the small cut-out box. When you slide the box toward the bot-

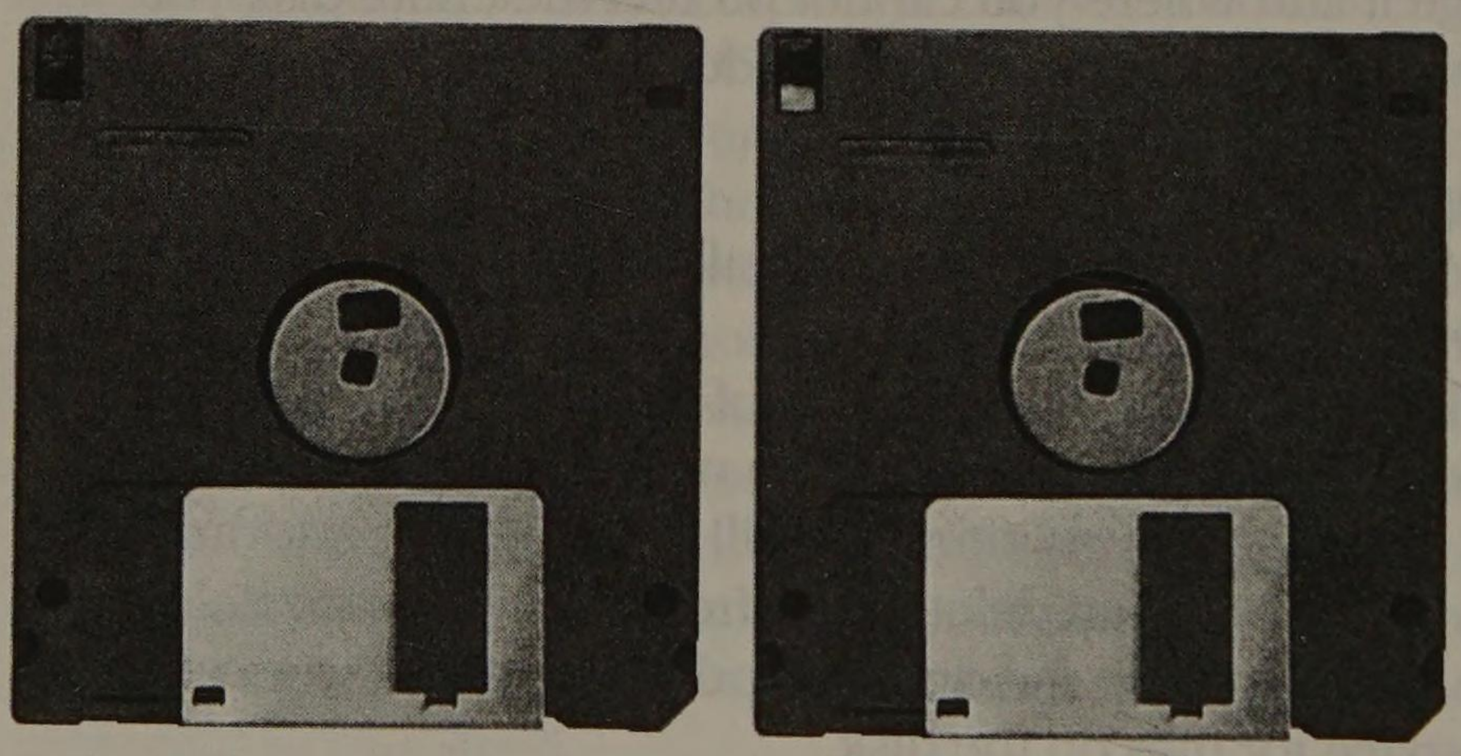
tom of the disk, the disk is write-protected. Figure 3-9 shows two 3 1/2 inch disks. The one on the left is unprotected and the one on the right is unprotected.

Figure 3-8: Unprotected and Protected 5 1/4 Inch Disk



The unprotected disk does not have a tab covering the notch.

Figure 3-9: Unprotected and Protected 3 1/2 Inch Disk



The unprotected disk has a tab slide to cover the hole in the disk.

Protecting an Individual File from Accidental Loss

The DOS ATTRIB command allows you to set the attributes of a file. Attributes control some of the tasks you can complete with files. One file attribute you can set is whether the file is available for read and write activities or only for read requests.

If you have a program file that you do not want users to accidentally delete you can set the file's attribute to read-only. If you have created a template or boiler-plate text in word processing that you want to make available for many users but you want to make sure they do not modify your file, set the file's attribute byte to read-only.

To set the file STATES in the current directory to read-only you would type:

```
ATTRIB STATES +r
```

If STATES were in the WP directory in drive D you would type:

```
ATTRIB D:\WP\STATES +r
```

The default setting for a file is to allow read and write activities. This means you will not have to use ATTRIB to enable write abilities for a file unless you previously turned off this capability with another ATTRIB command.

Copying an Entire Removable Disk

You can copy individual files on a file-by-file basis or you can copy the entire removable disk at once. When you elect to copy an entire disk of information, DOS writes the data on the second disk to make the disk identical to the source disk from which you are copying information. Any data that was on the target disk before the copy operation started will be destroyed because the disk copy operation writes over the existing data on the disk. Files on the source disk are not reorganized in any way. If the files are fragmented into many separate pieces this fragmentation will still exist on the new target disk since the copy operation occurs on a track-by-track, sector-by-sector basis rather than on a file basis. You must use either XCOPY *.* or COPY *.* if you want to copy all the files on a source disk to a target disk and also have DOS reorganize file fragments into one file.

Copying Between Disk Drives

When you have two removable disk drives with the same size and capacity you can use both drives to complete the copy operation quickly. To request that DOS make a copy of the source disk in drive A to the target disk in drive B, type:

```
DISKCOPY A: B:
```

If you are using a system without a hard disk, you will need to place your DOS disk in drive A before pressing ENTER since DISKCOPY is an external program and DOS will need to read the program from drive A before starting the copy operation. DOS prompts you to place the source and target disks in their respective drives and begins the copy operation. If it encounters an unformatted disk in the target drive, it formats the disk before it begins the copy operation.

If you only want to copy one side of a single sided disk you can enter DISKCOPY with a /1 switch as in:

```
DISKCOPY A: B: /1
```

Another switch option, /V, causes DOS to verify the copy operation. This adds time to the copy operation since DOS must read the data again and perform a comparison.

The only difference between the two disks when the DISKCOPY operation is complete should be the volume label on the two disks. DOS always assigns a new volume serial number to the disk in the target drive when the disk copy operation is complete.

Copying with a Single Disk

If you only have one drive to use in the copy operation, DISKCOPY can still handle the task. Because of the single drive you will need to swap the disk in and out of the drive. The procedure that DOS uses is to read as much as possible of the source disk into memory. Next, a message is displayed to indicate that the target disk should be placed in the drive. Once the target disk is in the drive and ENTER is pressed the data in memory is copied to the target diskette.

■ ***NOTE:*** It is a good idea to write-protect the source disk, especially when you are using a single disk drive. The write-protect tab will allow you to complete the procedure without worrying about placing the wrong disk in the drive.

You can continue copying disks by responding with a Y when DOS prompts you that the copy operation is complete and asks if you wish to continue copying.

If you do not specify drives for the DISKCOPY command, DOS assumes the current drive is the source and will prompt for the target drive except when the source drive is the fixed disk. When this occurs DOS displays an **Invalid drive specification message.**

Dealing with DISKCOPY Problems

If DOS displays a message telling you that the drive types are not compatible, you will have to repeat the DISKCOPY operation and specify the same drive for the source and target disk. You will also see error messages if you accidentally write-protect the target disk, use a damaged disk for the source disk, or attempt to use a target disk with bad areas on the disk. In the first of these problems you will need to "unprotect" the disk. If you experience a problem with the source disk you can use the RECOVER command in an attempt to reclaim as much as possible of the disk or use COPY *.* to copy as many individual files as possible. If the problem is with the target disk, you can always use another target disk.

Comparing the Contents of Two Disks

If you have two disks you think are identical, you can have DOS verify that there are no differences in the two disks. DOS is able to compare the disks on a track-by-track basis and insures, not only that the two disks contain the same files, but that the files contain the same data and the disk directories are also identical.

Entering the DISKCOMP Command

If you have two identical disk drives the quickest way to complete the compare operation is to place a disk in each disk drive and type:

```
DISKCOMP A: B:
```

If you are using a system without a hard disk you will need to have the DOS disk in drive A before pressing ENTER since DISKCOMP is an external DOS command. DOS will prompt for the insertion of the source and target disks before beginning the compare operation.

If you want the compare operation to look at just the first side of the disks you can add the /1 switch as in:

```
DISKCOMP A: B: /1
```

If you want the compare operation to look at the first eight sectors on each track you can use the /8 switch. Even if the disks have nine or 15 sectors only the first eight will be checked if you use this switch.

If you only have one drive to use for the compare operation, specify the same drive for both the source and target disks. If you specify only one drive, DOS ignores your entry and uses the default drive as both the source and target disk. DOS will prompt you when to change the source and target disks when you work with a single drive.

A compare operation which ends successfully will display the message **Compare OK**. DOS will then prompt to see if you wish to continue comparing disks.

Dealing with Comparison Problems

There are a number of factors that can cause problems with DISKCOMP. If you attempt to use two incompatible drives, DOS will display a message informing you that the drives are not compatible. You will need to use an entry such as:

```
DISKCOMP A: A:
```

and swap the source and target disk in and out as needed.

DISKCOMP will not work on network drives. Drives that you have used with ASSIGN, JOIN, and SUBST are also not acceptable.

DOS returns error codes that indicate the specific problem with the comparison. If you wish to perform the compare from within a batch file you can obtain additional information on the failure of the compare operation.

At a Glance

Disks are your most important resource for working with your computer. Even if your hardware system is destroyed, a copy of your important data on disk will allow you to pick up exactly where you left off with a new computer. The following chapters will provide more detail about the DOS commands you can use to manage your disk information. Some information you have learned about disks in this chapter are:

- **Disks that look the same may have different capacities**
- **Disks are organized with tracks and sectors**
- **Disks can have multiple directories, partitions, and logical drives**

- **Protect a file on a disk—ATTRIB +r**
- **Check the capacity of a disk—CHKDSK**
- **Copy all the files on a disk to another disk—DISKCOPY**
- **Compare two disks to insure that they are identical—DISKCOMP**

Chapter 4

Using the Shell for File Management Tasks

In Chapter 1 you were introduced to the two different ways you can interface with DOS. Although Chapters 2 and 3 focused on concepts and basic tasks completed at the DOS prompt, you will want to turn your attention to the DOS Shell again. While it does not provide access to the full set of DOS options, the DOS Shell offers time-savings and an easy-to-use alternative for the tasks it can handle. Instead of typing filenames you can use your mouse or keyboard with the Shell to select one or more files that you want to work with. You can perform DOS tasks such as copying, renaming, or deleting files with the Shell without memorizing the syntax of commands that would be needed for an entry at the DOS prompt. You will also find that you can easily customize the Shell appearance, run programs, and work with data on all disks and directories. The Shell does not provide access to the full range of DOS commands, however the options that are available are so easy to use, even a first-time DOS user can be assured of success.

What You'll Learn

In this chapter, you will learn how to:

- **Identify different parts of the DOS Shell screen**
- **Make selections in the Shell screen and menus**
- **Change the appearance of the files on the screen**
- **Display help information on a topic**
- **Select files**
- **Perform file commands such as copying, moving, renaming, deleting, and changing attributes**
- **Run a program**

- **Create and remove directories**
- **Change DOS Shell settings such as screen color and confirmation messages**

Loading the Shell and Exiting to the DOS Prompt

If you use the default installation procedure, the DOS Shell will not appear when you load DOS. You can change this default to make the DOS Shell appear each time DOS is loaded.

Whichever way your system is set, switching between the DOS prompt and Shell is quite easy. If the DOS Shell does not appear on your screen, type **DOSSHELL** and press ENTER to switch to the Shell. If you are first using a floppy disk system (no hard disk), you must put the SHELL disk in drive A before typing DOSSHELL. The DOS Shell will remain on the screen. However, you can temporarily exit the Shell by pressing SHIFT-F9, or permanently leave the Shell by pressing ALT-F4 or selecting Exit from the File menu. If you temporarily leave the DOS Shell, you can return to it by typing **EXIT** and pressing ENTER. When you return after a temporary exit, everything will be exactly as you left it.

Exploring the Shell Screens

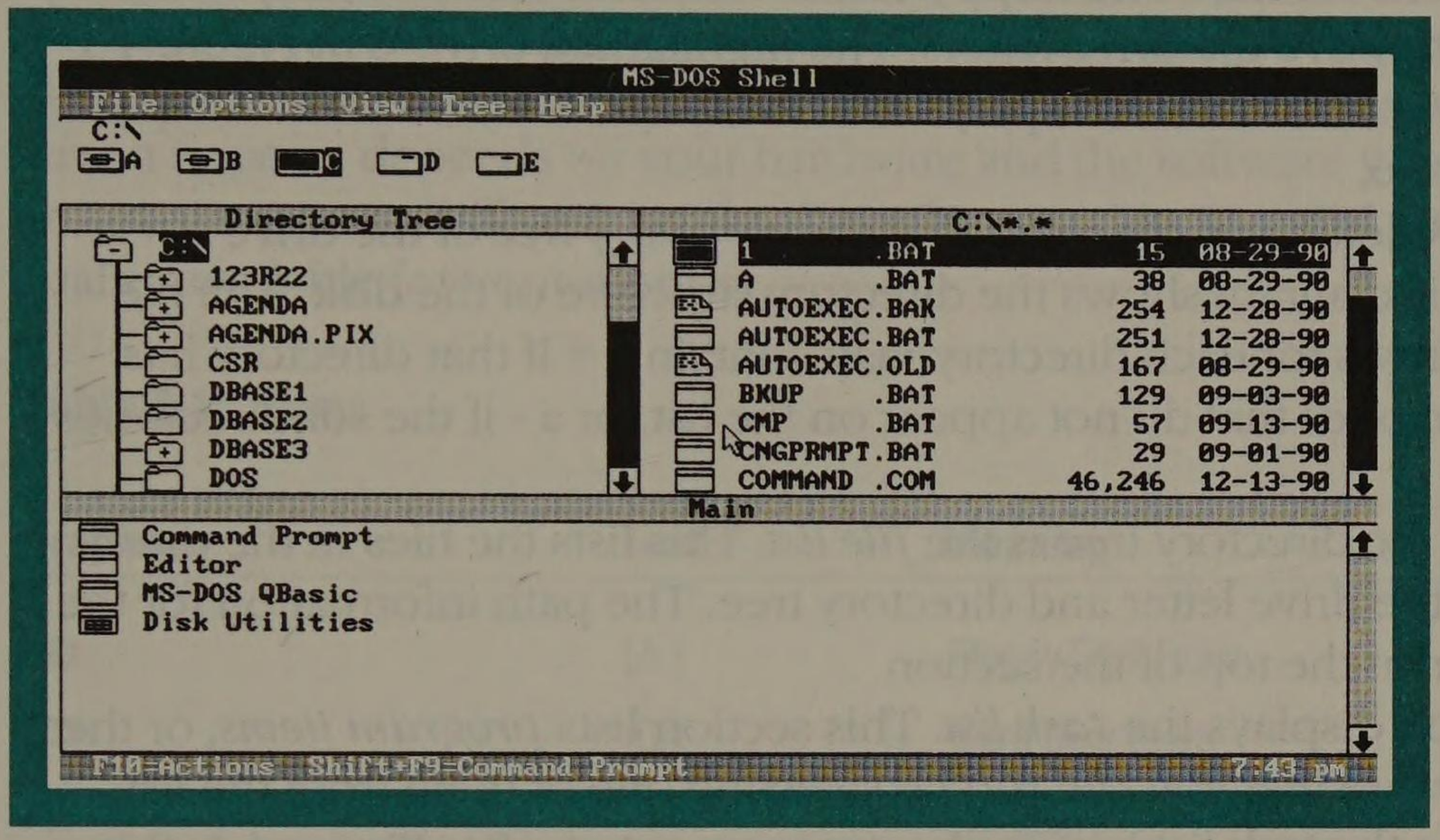
If you have been using DOS with DOS command prompt entries, you will be surprised at the difference in the screen when you work with the DOS Shell. The DOS Shell screen is much more like your other application programs and it effectively makes use of the entire screen. The DOS Shell screen is divided into several sections to display the available drives, the directory structure of a drive, the files in a directory, and the programs you can run. You will want to become familiar with the parts of the DOS Shell screen so you can use it effectively as you learn about the different tasks you can perform with the Shell. Your screen will look different depending on whether you are using a text or graphics screen mode. Figure 4-1 shows a graphics DOS Shell screen and Figure 4-2 shows a text graphics DOS Shell screen.

Understanding Screen Elements

The DOS Shell screen contains different elements that present the menu, the files, and the programs you can use. The first line is the *title bar* which displays the name of the Shell. Below the title bar is the *menu bar* which contains the names of the available menus. When you select one of these menus, the DOS Shell

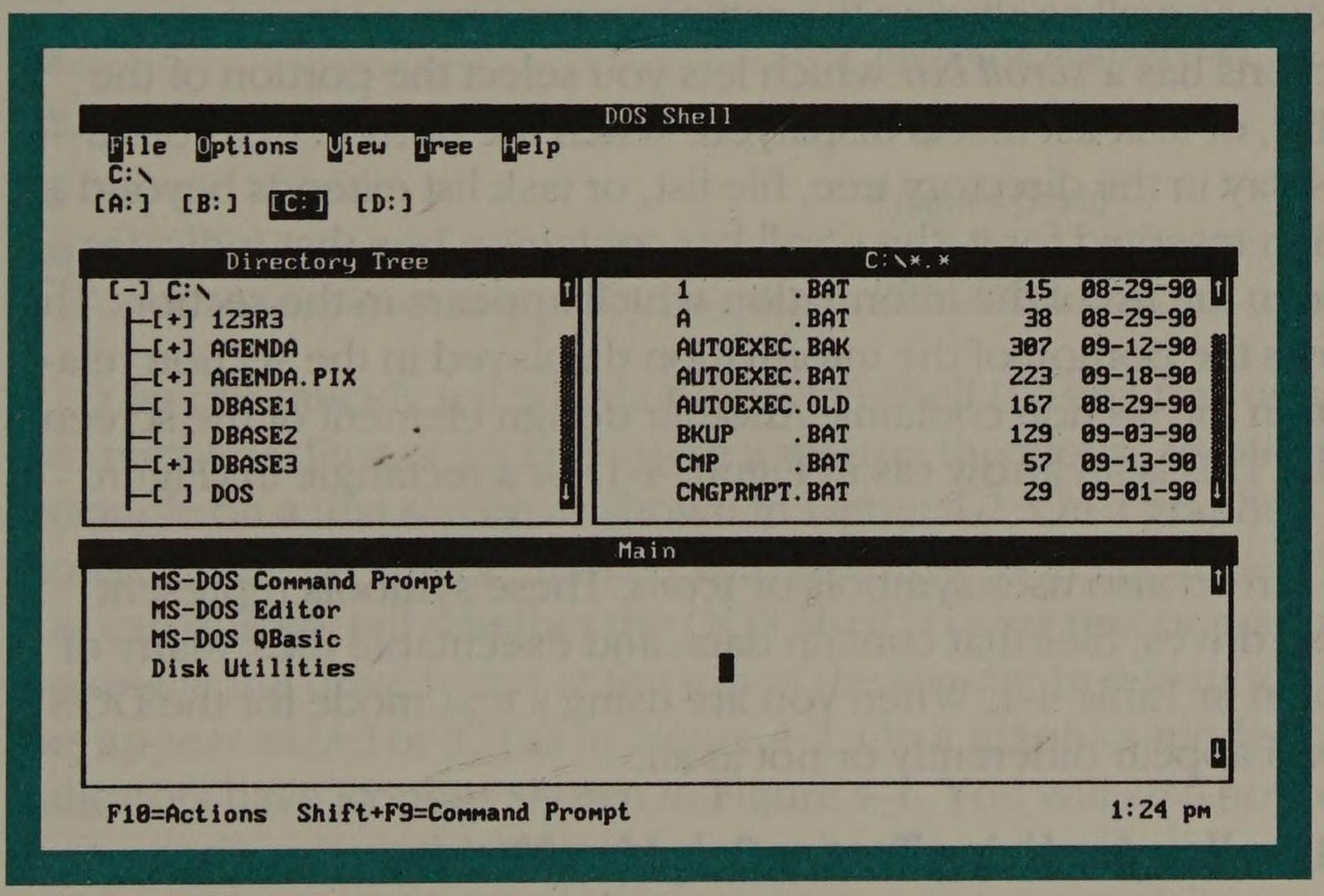
displays the menu associated with the menu name you select. The bottom line is the *status line* which displays lists of keys you can press or other information.

Figure 4-1: DOS Shell Screen Using Graphics Mode



Graphic images represent disk drives, files, and programs.

Figure 4-2: Text Mode Display of DOS Shell Screen



Without graphic support, only text entries display on the screen.

The rest of the screen below the menu bar is the *view*. The top portion of the view displays the drives that are available. As you can see from the graphics screen in Figure 4-1, the DOS Shell displays different symbols for floppy disks (drives A and B) than for hard disks (drives C, D, and E). Text screens do not have different designations for hard and floppy disks. The drive and directory currently selected are shown above the drive letters. The highlighted drive C in Figure 4-1 shows the *selection cursor* which displays the currently selected item in the part of the view that is active.

Below the drive letters on the left half is the *directory tree* of the drive currently selected. This diagram shows the directory structure of the disk. The file folders that DOS draws for each directory may contain a + if that directory has additional subdirectories that do not appear on the list, or a - if the subdirectories are displayed.

To the right of the directory tree is the *file list*. This lists the files in the directory selected with the drive letter and directory tree. The path information for the listed files is shown at the top of the section.

The final section displays the *task list*. This section lists *program items*, or the programs and software that you can run by selecting one of them from the list. Each of the items in the task list is part of a *program group*. (For Figure 4-1, this group is called Main.) DOS also includes another program group called DOS Utilities, but you can create your own. You can also add and delete programs to and from a program group as well as altering the settings a program uses.

Each of the sections has a *scroll bar* which lets you select the portion of the directory tree, file list, or task list that is displayed. When the portion of information DOS has to display in the directory tree, file list, or task list extends beyond the size of the section reserved for it, the scroll bar contains a box that indicates the relative position in the list of the information which appears in the section. The size of the box shows the portion of the information displayed in the section relative to the information the section contains. Another design element of the screen is the *mouse pointer*. This is an arrow (as in Figure 4-1) or a rectangle in (Figure 4-2).

The DOS Shell screen also uses symbols or *icons*. These symbols represent hard and floppy disk drives, files that contain data, and executable files. Many of these icons are shown in Table 4-1. When you are using a text mode for the DOS Shell, these icons will appear differently or not at all.

Determining Whether You Are Using Text or Graphics Mode

The DOS Shell can operate in text or graphics mode if your monitor supports graphics. The text mode is quicker than graphics but it does not look as nice. To

get an idea of the difference between the two modes, think of the difference between using a piece of paper in a typewriter or drawing on the paper with a pencil. A typewriter, like a computer's text mode screen, can display a limited set of characters in specific locations. For a text screen, it is 80 columns across and 25, 43, or 50 lines down. When you draw something on a piece of paper, you can draw anything on the screen but it will take you longer than using a typewriter. Also, just as your drawing skill may vary from someone else's, the way a graphics screen appears depends on your hardware and the software you use. The Shell program supports different levels of graphics quality. Usually, DOS uses the best quality available for your monitor.

Table 4-1: Icons

Graphics	Text	Meaning
	[A:]	Floppy Disk Drive
	[C:]	Hard Disk Drive
	[ ]	Directory
		Data and Non-executable Files
		Executable Files
		Moving or Copying Files
		Program
		Program Group
RAM	[E:]	RAM Disk

You can quickly tell whether the DOS Shell is using text or graphics mode by two methods. First, if you are using a mouse, the mouse location is marked by a rectangle on a text screen as shown in Figure 4-2. On a graphics mode screen, the mouse's position will be marked with an arrow as in Figure 4-1. The other way you can quickly tell whether the DOS Shell is using text or graphics mode is by looking at the drive letters at the top of the screen. In text mode, these drive indicators appear as [A:] or [C:] as in Figure 4-2. On a graphics mode screen, these drive indicators have icons as shown in Figure 4-1. You will also notice other differences, such as how the lines are drawn between directories and files.

Interfacing with the Shell

In Chapter 1 you learned the basics about interfacing with the DOS Shell. Here you will have an opportunity to review some of the basics and learn new techniques. To use the Shell effectively, the first thing you need to master is making selections. The method you use to make a selection depends on whether you are using the mouse or the keyboard. You will also want to become familiar with making menu selections or backing out of them when you change your mind. You must also be able to select menu commands and be familiar with the types of information that the DOS commands will expect.

Learning to Use the Keyboard and Mouse

You will use either the keyboard or the mouse to make your selections. Using the mouse allows you to quickly make selections by pointing to and clicking on what you want. You can make many of your selections using the mouse. However, in some situations you may also need to use the keyboard to provide characters needed for menu selections. If you do not have a mouse, it will be necessary to use the keyboard.

Using a Keyboard You can use the keyboard to make all of your selections in the DOS Shell. A keyboard looks like the one in Figure 4-3. This shows the enhanced keyboard that most AT and higher level machines use, although some of the keys may be in different locations.

Figure 4-3: Example of a Keyboard

Note the separate cursor movement keys.

At the top of the keyboard are the *function keys*, which are labeled F1 through F12 (although some older keyboards only have up to F10). These function keys are assigned to different tasks by the software you are using. The way DOS uses them will be described with the features that use these keys. The function keys will take on different meanings when you use other software, since each software package has the option of assigning different functions to each function key. This allows a software developer to make frequently-used operations available by the touch of a key.

On the right side of the keyboard are the *movement keys*. These keys move a cursor or highlight to another direction. The newer keyboards have a second set of arrow keys and other movement keys separate from the numeric keypad. The numeric keypad on the extreme right lets you switch between entering numbers and using the movement keys by pressing the key labeled NUM LOCK. Older keyboards will not have the second set of movement keys.

The main section of the keyboard is very similar to typewriter keys though some of the keys are different. The CAPS LOCK key which a typewriter uses to type the second set of characters, normally the uppercase letters and special symbols, only operates on letters. This means while CAPS LOCK is active, the letters you type are uppercase and the numbers you type are the numbers rather than the symbols. You can change the case for individual letters and type the symbols above the numbers by holding down the SHIFT key as you type the letter or number. The ESC, CTRL, and ALT keys are unique to computers. ESC removes the current entry or backs you out of a menu setting. It allows you to escape from the activity you are currently performing. The CTRL and ALT keys are designed to be used in combination with other keys such as CTRL-R, which means that you hold down the CTRL key while you type an R. Programs can use these keys to provide additional functions. The DOS Shell uses the combination of ALT and letters of the alphabet to select menu options. For example, pressing ALT-F gives you a quick approach for opening the menu and selecting File—all in one step.

When you use the keyboard in the DOS Shell, you can switch from section to section by pressing TAB or SHIFT-TAB. TAB moves to the next section and SHIFT-TAB moves to the previous section. You can move between the items listed in each section using the arrow keys. As you move to a different location in the files list or to another section, DOS updates your position in the list on the scroll bar. You can use PGUP and PGDN to page through the list. Press HOME or END to scroll to the beginning or end of the line of an item in the list. You can also use CTRL-HOME and CTRL-END to scroll to the top or bottom of the list.

Using a Mouse A mouse allows you to make selections by pointing and selecting what you want. To use a mouse, hold the mouse in the palm of your hand. This lets you use your fingers to press the mouse buttons while the palm of your hand moves the mouse. As you move the mouse, the mouse pointer will move across the screen. Most mice can be used on any flat surface although some require a special pad. Mice can contain one or more buttons. The DOS Shell only uses the left-most button to make selections, although you can switch to use the right-most button. The DOS Shell only uses one button and ignores any others. To select an item, such as selecting a file to use, point to it and click it by pressing the left button. To select an item and complete the selection process, such as selecting an item in the task list, double click the item by pressing the left button twice quickly.

When you use the mouse in the DOS Shell, you can switch between sections by pointing to the one you want and clicking on it. You can also change the part of the list that appears in a section by moving the box in the scroll bar. If you click the scroll bar above the box's current position, DOS scrolls the list to display the items in the list above the ones currently shown. If you click the scroll bar below the box's current position, DOS scrolls the list to display the items in the list below the ones currently shown. You can also click the arrows at the ends of the scroll bar to scroll the display one line at a time. Drag the box on the scroll bar to a new location and DOS will display the items that appear when the scroll box is in the new position. *Dragging* a mouse means moving to the starting point and holding down the button on the mouse as you move to the ending point.

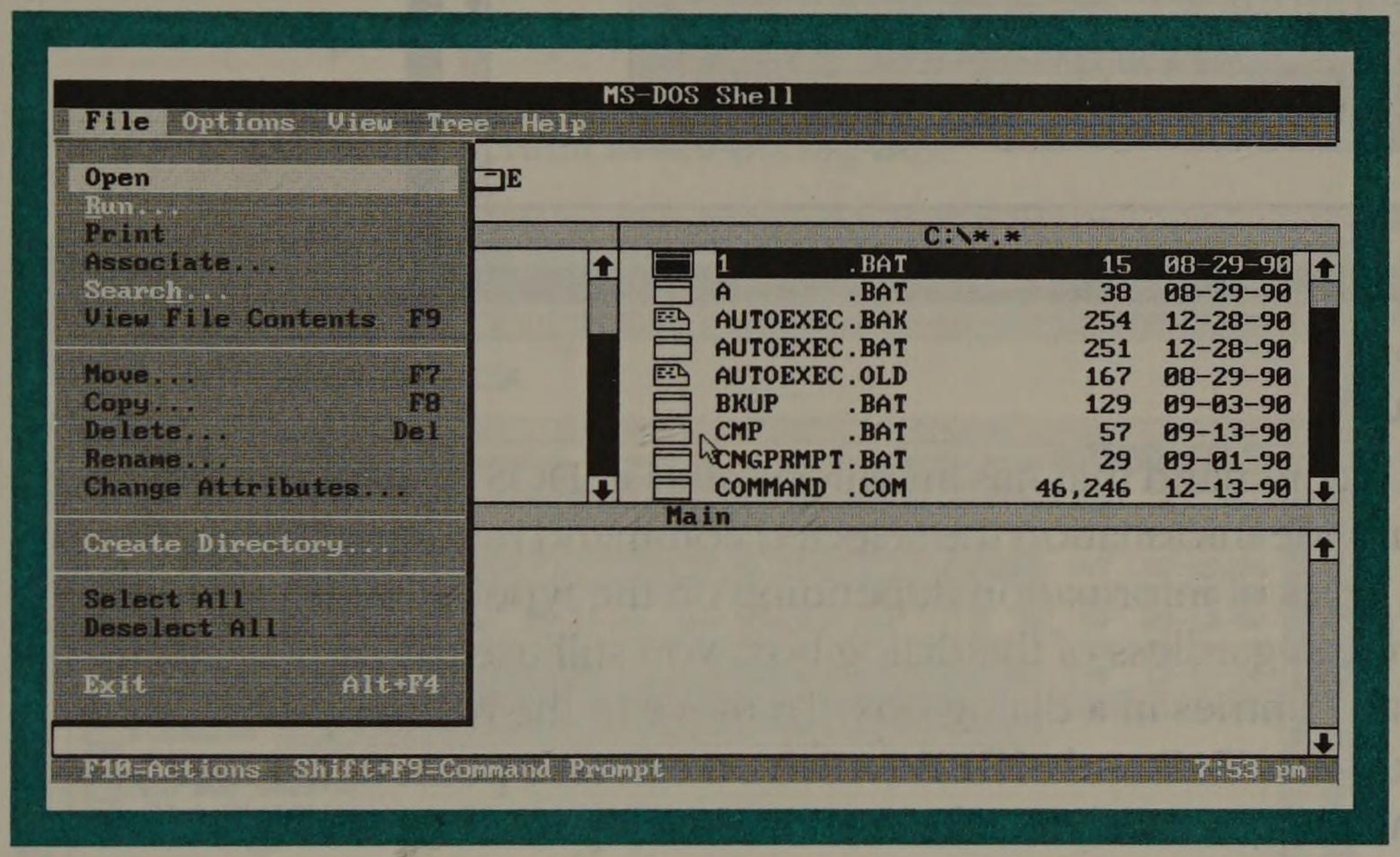
Making Menu Selections

Menu selections let you perform a DOS command quickly. The first step in using the menu is activating it. You can activate the menu with a mouse by pointing to the menu you want and clicking the mouse. With a keyboard, press F10 or the ALT key once and type the underlined letter or the letter displayed in a different color of the menu that you want. This displays a menu like the File menu shown in Figure 4-4. From the menu, select the command you want by pointing to it and clicking on it, typing the underlined letter of the one you want, or using the arrow keys to highlight the one you want and pressing ENTER. If one of the items is dimmed, you cannot select it. You may also see an ellipsis (...), which indicates that DOS needs more information to carry out the requested task. If you select the command, you will see a *dialog box* like the one for the File Display Options command in the Options menu shown in Figure 4-5. You will use this box to provide specific command information which allows DOS to complete your request. Another possibility is a diamond next to the menu item. A diamond

indicates that the menu item was selected and is still in effect. The last type of text you may see next to a menu item is a key combination which you can press in place of selecting the command. For example, instead of using the Copy command in Figure 4-4, you can press F8. Unless a command contains an ellipsis, DOS will perform the command for the menu item when you select it. In this chapter, when the text says **select or enter a command** it means that you activate the menu and select the menu to use one of the commands from that menu.

- ***NOTE:*** Initially, you will want to make your selections from the menu, but in no time you will learn the accelerator key sequences that are available, enabling you to quickly make your selections with confidence.

Figure 4-4: File Menu



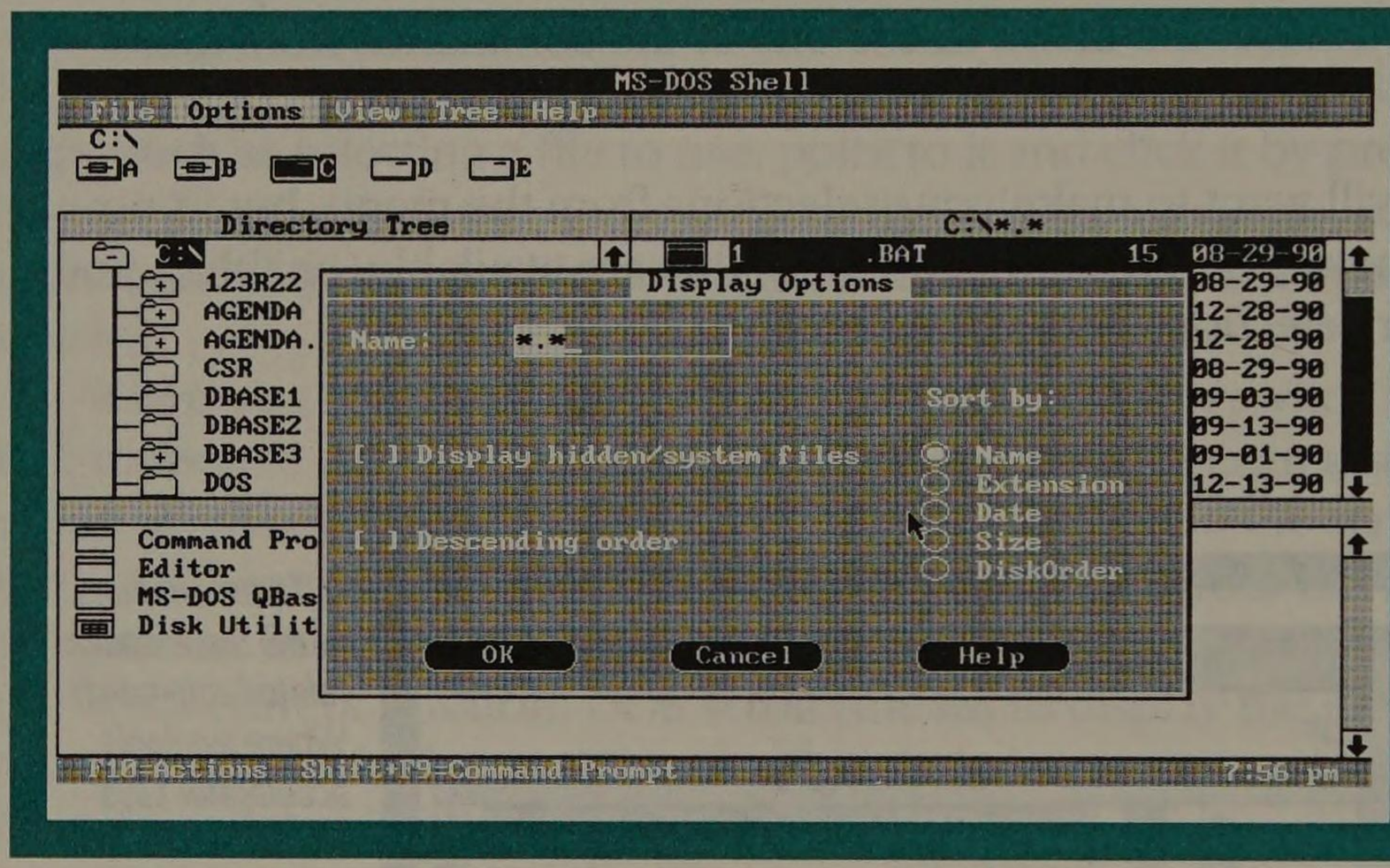
Different sections of the menu show related commands. Where available, accelerator keys are shown on the right side.

Undoing a Selection

After you make a menu selection, you may decide you do not want to perform the command after all. You have several choices for backing out of a menu command you are selecting. With a mouse, you can cancel a command by clicking another part of the screen. You can also press ESC to leave the menu command and deactivate the menu bar. A third option is to press ALT or F10, which leaves the menu bar but keeps it active so you can select a menu option. If you are in a

dialog box, you can leave the dialog box without performing the command by pressing ESC or using the Cancel command button that is described below.

Figure 4-5: File Display Options Dialog Box from the Options Menu



**.* entry can be replaced with any valid pattern for a filename.*

Using Dialog Boxes

When you select a command that has an ellipsis after it, DOS displays a dialog box which lets you provide information the selected command requires. Dialog boxes contain different types of information depending on the type of information the command requires. Regardless of the dialog box, you still use the same steps to make selections and entries in a dialog box. To move to the next or previous part of the dialog box, press TAB and SHIFT-TAB. You can also point to the area you want and click the text you wish to select. Within each part of the dialog box, if it contains several selections, you can use the arrow keys to make selections.

At the bottom of the dialog box are *command buttons*. These command buttons usually include OK to finish the command, Cancel to abandon the command and Help to display Help information about the command. A command button is selected when it contains an underline.

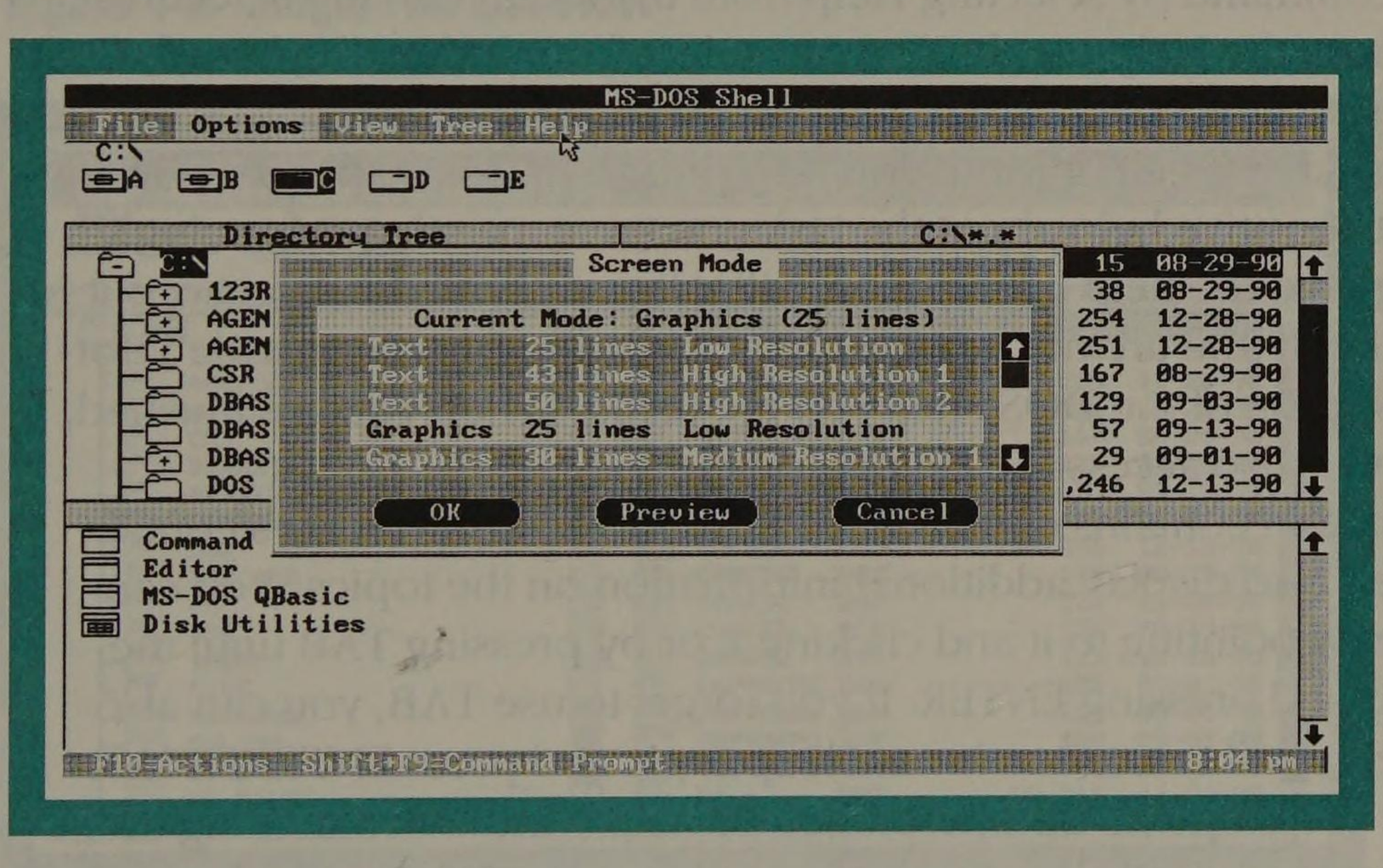
Another type of information you may need to provide is text, such as a filename or program description. A dialog box that requires text displays a rectangle like the Name text prompt in Figure 4-5. If the rectangle already contains text,

you can replace it by making a new entry or edit it by pressing the RIGHT ARROW and then moving to where you want to change.

Dialog boxes may also contain *lists* that list possible selections you may choose. These lists have a scroll bar you can use with a mouse to change the section of the list that appears. In the Screen Mode dialog box in Figure 4-6, the display options are in a list. You would select the one you want by clicking it or using the arrow keys to highlight the one you want.

The remaining type of selection you can make in a dialog box are *option buttons*. You can select as many options as you want, as in the ones shown in Figure 4-5. These option buttons contain an X when selected and have square brackets. To select or deselect one of these option buttons, point to it and press the space bar or click the mouse. The other type of option buttons are mutually exclusive options such as the sort order buttons in Figure 4-6. The currently-selected option has a dot and the other options have parentheses. When you select one of these it replaces the current selection. To select or deselect one of these option buttons, point to it and press the space bar or click the mouse.

Figure 4-6: List in the Screen Mode Dialog Box

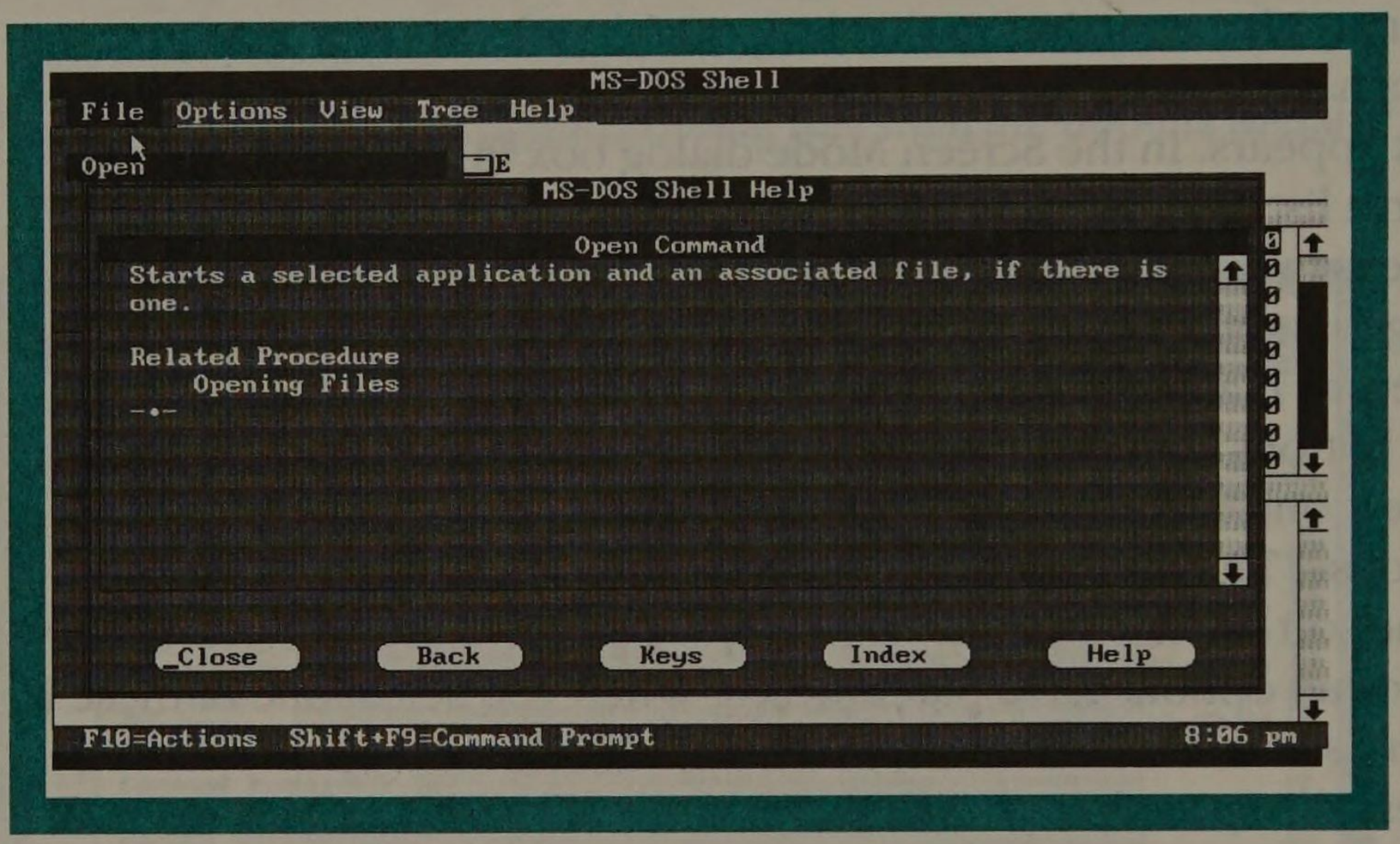


You can use the mouse or keyboard to select a different screen mode.

Getting Help

The DOS Shell provides access to an extensive help system that will supply information on every aspect of the DOS Shell. You can display general information about the DOS Shell or specific information about a particular menu choice. Figure 4-7 shows a help window for the Open command in the File menu.

Figure 4-7: Help Window for the Open Command in the File Menu



Help buttons at the bottom provide different options.

Another way to display help information is to use the Help command. You can execute the Help command by selecting Help from the menu bar and selecting one of the topics from the Help menu. You can select from Index, Keyboard, Commands, Procedures, Shell Basics, Using Help, and About Shell.

One way of displaying help information is to press F1. This displays a window containing context-sensitive help about the task you are currently performing. If you are in a dialog box, the help information will tell you about the selections it expects you to make. If you are not using a menu, pressing F1 will provide information about the section of the DOS Shell you are using. When you are finished, press ESC. ALT-H will also access the Help information.

All help information contains common features. Most help windows contain buttons you can select to display additional information on the topics. You can select one of these by pointing to it and clicking it or by pressing TAB until the topic is highlighted and pressing ENTER. If you forget to use TAB, you can also press the arrow keys until the cursor is on a help topic and press ENTER. When you select another help topic, DOS opens another help window containing the help information for the selected topic. Most help windows contain five buttons—Close, Back, OK, Keys, Index, and Help. Selecting Close exits Help and closes the current help window. Selecting Back redisplays the previous help topic or takes you back to where you were before requesting help. Selecting Keys displays the initial help window for using different keys in the DOS Shell. Selecting Index displays the index of help topics. Selecting Help displays information about using

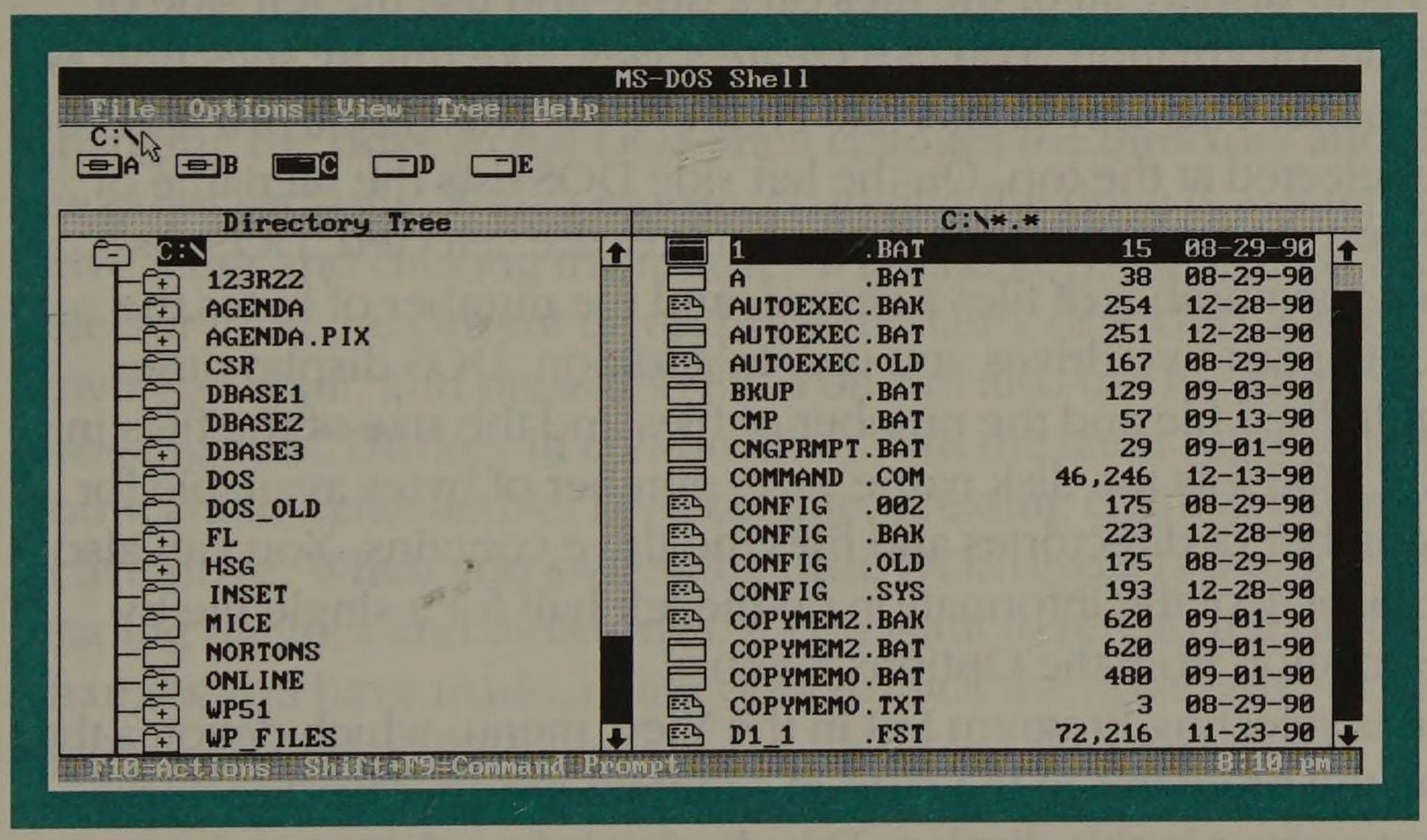
Help. Besides returning to a previous help window by selecting the OK button, you can also display a previously displayed help window by clicking part of the help window.

Changing the View

The DOS Shell has several different options for how directories, files, and programs appear in the Shell. While the default splits the screen between the directory and files and the tasks, you can select another view to display directories and files for two disks; directories and files for one disk; all files and the information about the drive, directory, and selected files; or program groups. Change the view when you want to focus on the files of one or two directories, to review all the files, or to focus on the programs you can run from the task list. You can always return to the default by selecting Program/File Lists from the View menu.

To change the view so the directory tree and file list occupies the screen, select Single File List from the View menu. The view looks like Figure 4-8. This view uses the screen to display a larger directory tree and file list.

Figure 4-8: Single File List View

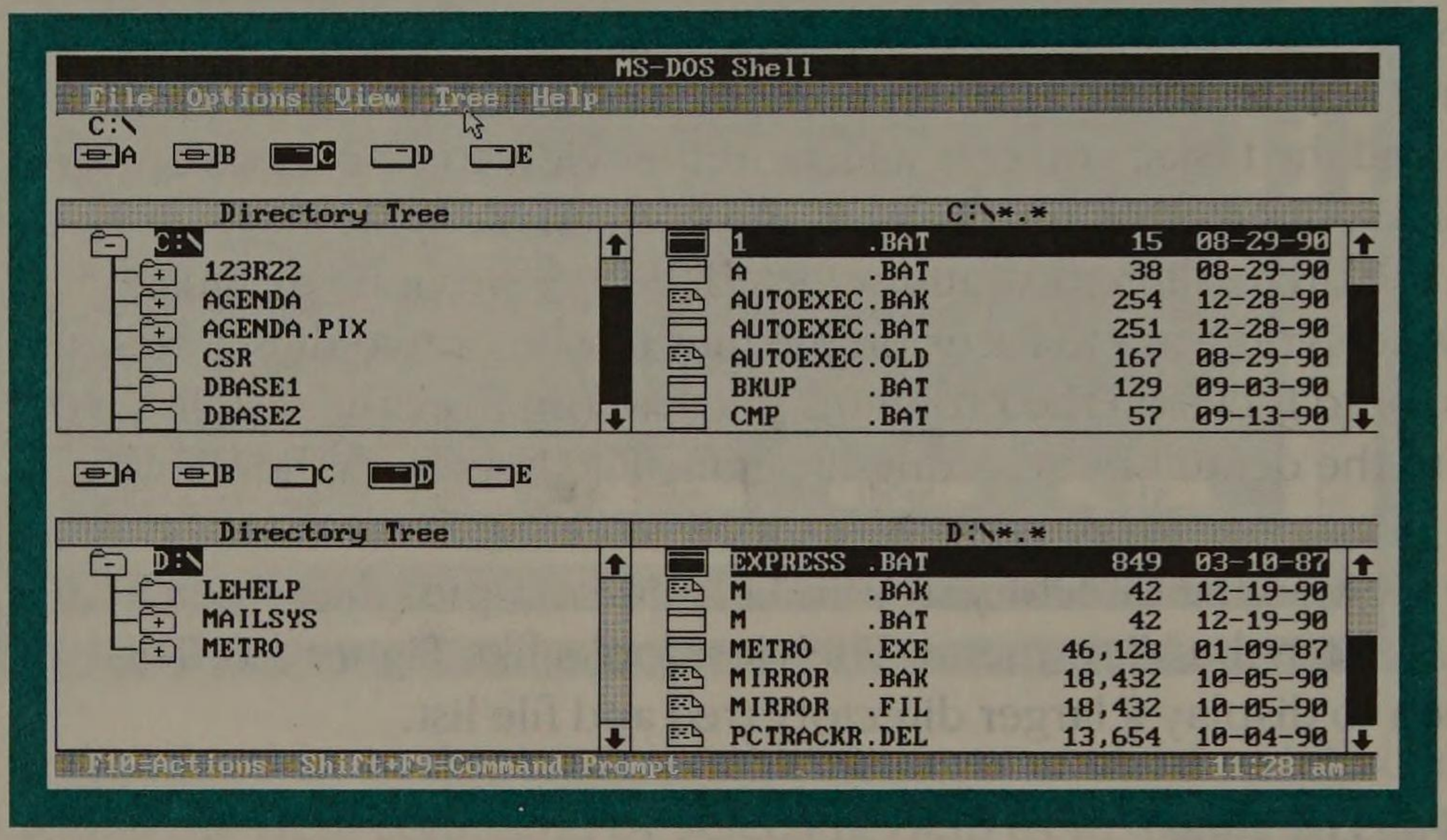


Single file list allows you to view directory tree and files.

You can also display the directory tree and file list for two disks simultaneously by selecting Dual File List from the View menu. This view looks like Figure 4-9. While the top and bottom half of the view are initially set to the same drive and directory, you can change them without affecting the other. You can use

this view to display the directories and files of two disks or the same disk and display the files for separate directories.

Figure 4-9: Dual File List View



You can look at two drives or directories with a dual file list.

A fourth option is to display all of the files on a drive and use the left side of the view to display file information. You can create a view like this by selecting All Files from the Tree menu. This view looks like Figure 4-10. This displays a list of all files in the drive selected at the top. On the left side DOS lists the filename of the highlighted file and any attributes the file has. If files are selected, DOS displays the drive letter, the number of files selected, and the number of bytes the selected files use for the last two drives you use. In addition, DOS displays the directory of the highlighted file and the number of files and the size of the files in the directory. Finally DOS lists the disk name, size, number of bytes available for new files, and the number of directories and files the drive contains. You can also create a temporary display of the information on the left half for a single file by selecting Show Information from the Options menu.

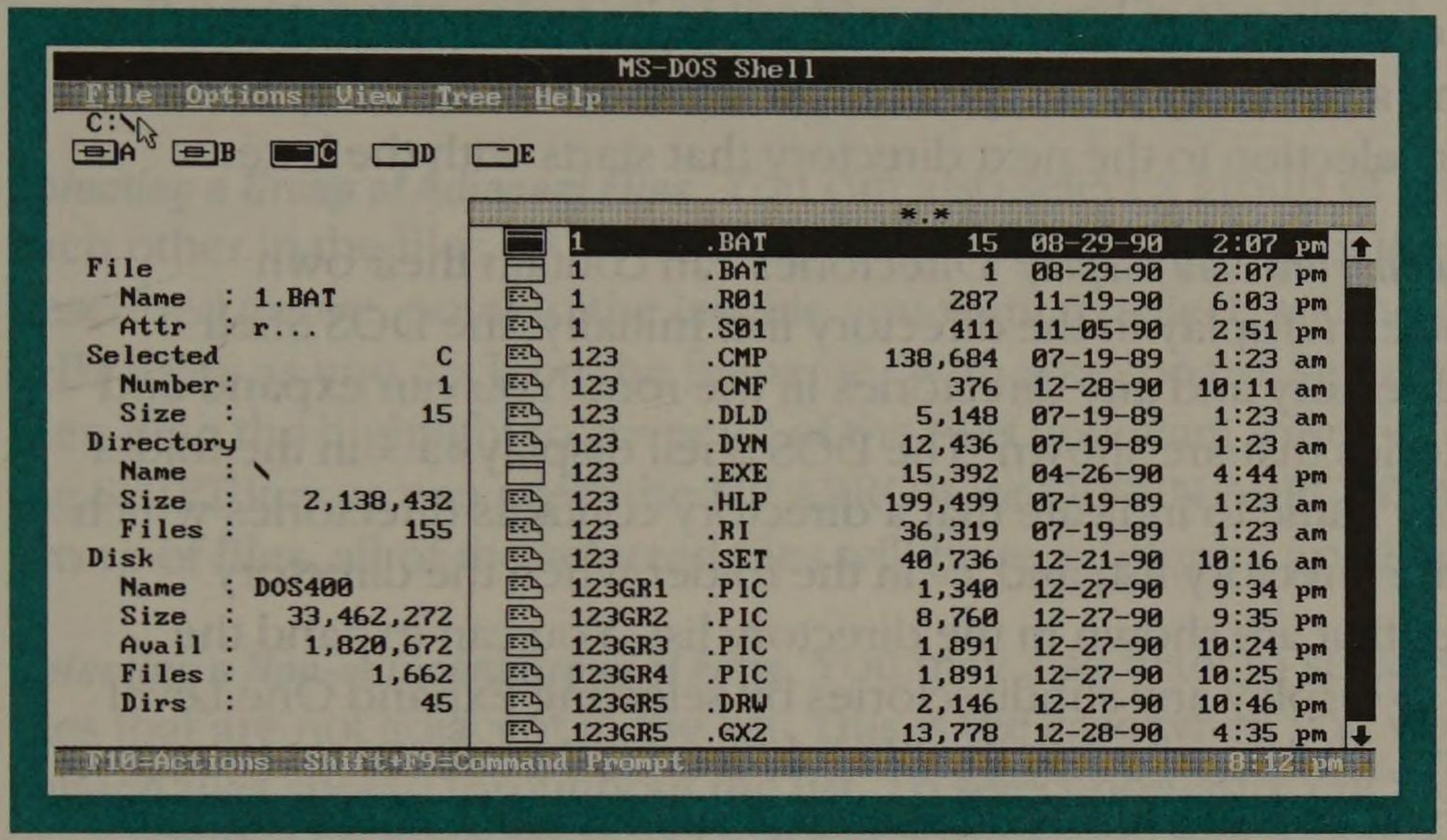
The final option is selecting Program List in the View menu, which devotes the entire view to listing the programs in the current program group. The directory tree and files will not appear in this display. This display is for when you do not want to manipulate files using the file commands described later in the chapter.

Exploring the File Manager Screen

Most of the Shell view focuses on allowing you to manipulate files. You can change the drive, directory, and files listed. You can select files to work with and

the file commands you want to use. These file commands include copying, deleting, and moving files and all are covered in the Selecting File Commands section later in the chapter.

Figure 4-10: All Files View



Additional information about the file and disk are displayed to the left.

Changing the Drive

Changing the drive in the DOS Shell changes the directory and files displayed in the other sections. With a mouse, changing the drive is as simple as pointing to the drive letter and clicking the mouse. With the keyboard, press TAB until the selector is on the current drive, press the LEFT or RIGHT ARROW to highlight the drive you want, and press ENTER. You can also use the shortcut approach of pressing the CTRL key in combination with the letter representing the drive that you want to make active. For example, pressing CTRL-D will make drive D the active drive. When you switch the disk, DOS displays the box telling you that it is reading the files and directories. If you want to reread the disk to reflect any changes you have made, either double click the drive letter or select the drive again and press ENTER.

Changing the Directory

Once the drive is selected, you may want to change the directory that is listed. You may also want to expand or contract the directory listings. The selected directory determines the files listed in the files list. To select a different directory with a mouse, point to the directory name and click it. To select a different

directory with the keyboard, press TAB or SHIFT-TAB until the directory list is the active section, use the arrow keys to highlight the directory you want, and press ENTER. You can use the UP ARROW and DOWN ARROW to move to the previous or next directory. You can use CTRL-UP ARROW and CTRL-DOWN ARROW to move to the previous or next directory on the same level. Use the LEFT ARROW or RIGHT ARROW to move to the parent or subdirectory. Press HOME to move to the root directory or END to move to the last directory in the list. Another quick method of selecting a directory is to type the first letter of the directory name which moves your selection to the next directory that starts with the letter.

Collapsing and Expanding the Tree Display Directories can contain their own directories that you can display in the directory list. Initially, the DOS Shell displays the root directory and any directories in the root. You can expand and contract the directories that are shown. The DOS Shell displays a + in the folder next to the directory name to indicate that a directory contains directories which do not appear in the directory list, and a - in the folder when the directory contains directories that are shown in the directory list. You can expand the selected directory to display any subdirectories by selecting Expand One Level from the Tree menu, clicking the + in the file folder with the mouse, or by typing + while the Directory Tree is active. To expand all subdirectories below the current directory, select Expand Branch from the Tree menu or type *. To expand all subdirectories in the current drive, select Expand All from the Tree menu and press CTRL-*. You can contract the display to hide all the subdirectories of the current directory by selecting Collapse Branch from the Tree menu, clicking the - in the file folder with the mouse, or typing -.

Selecting Files

The DOS Shell requires that you first select the files you want to use. The steps you perform to select one or more files to work with depend on whether you want to select one file or several files. When selecting several files, the steps you perform depends on whether the files are next to each other in the list or whether they are in different parts of the list or in different subdirectories. When a file is selected, it has a triangle next to the filename.

Selecting a Single File To select one file to use with the mouse, point to the file in the list and click on it. If you are using a keyboard, you must press TAB until the file list is highlighted then use the UP ARROW and DOWN ARROW to highlight the file you want in the list. You can quickly move the highlight to the top or bottom file in the list by pressing HOME or END. A quick alternative to selecting a file, is to type the first letter of the filename. When you type a letter in one of the

sections, DOS moves the highlight to the next item in the list that starts with the letter. This also applies to directory names and tasks but not drives.

Selecting All the Files in the List At times you'll want to work with every file in a directory. You might want to copy all of the files to a disk or delete all of them. Selecting the entire group is quicker than working with the files on an individual basis. If you want to select all of the files displayed in the file list, select the Select All command from the File menu or press CTRL-/ (slash).

Selecting a Group of Adjacent Files You can also select a group of files that are next to each other in the files list. With a mouse, select the first file of the group as described above, point to the last file you want to select, and hold down the SHIFT key as you click on the filename. With a keyboard, select the first file and then drag the highlight to cover all of the files you want to select by holding down the SHIFT key as you press the UP ARROW or DOWN ARROW. As you highlight a group of files, all of the selected files will have a different appearance.

Selecting a Non-adjacent Group of Files You may also want to select several different files that are not adjacent in the list. This is the case when you want to work with several files spread throughout the list. To select non-adjacent files with a mouse, press CTRL while you click on each file. This means that if you want to select the 123.EXE and the 123.PAL files in Figure 4-10, you would point to the 123.EXE file and hold down the CTRL key while you clicked on it. Then you would point to the 123.PAL file and hold down the CTRL key while you clicked on it. To select non-adjacent files with the keyboard, press SHIFT-F8 to display Add in the status line. This adds the files you select to the group rather than assuming that the files you select are replacing the files you selected previously. Then highlight each file and press the space bar. If you decide you want to start over, press SHIFT-F8 twice to remove Add and then start over.

Selecting Several Groups of Files A fourth option is to select several groups of files from the file list. To do this with a mouse, select the first group of adjacent files using the SHIFT key as described above. For the remaining groups of adjacent files, hold down the CTRL key while you click on the first file of the group and then hold down the CTRL and the SHIFT keys while you click on the last file of the group. To do this with a keyboard, select the first group of adjacent files using the SHIFT key and the arrow keys as described above. Press SHIFT-F8 and then for each group of adjacent files, highlight the first file in the group, press the space bar, and select the other adjacent files by holding down the SHIFT key as you press the UP ARROW and DOWN ARROW keys.

Selecting Files in Different Directories If you are selecting files in several directories, you will need to use the Select Across Directories command in the Options directory. This is used when you are selecting files from different directories that you want to copy or move to a new location. If you do not use this command, DOS only remembers the selected files from the last directory in which you were working. When the diamond appears next to Select Across Directories, DOS continues to remember all of the files you have selected. You can add or remove the diamond next to this command by selecting Select Across Directories.

You may later realize that you have selected a file you did not want to include in the batch. You can deselect a file. With a mouse, point to the file to deselect and hold down the CTRL key while you click on the filename. With a keyboard, highlight the file and press CTRL-space bar. If you want to deselect all of the files you have chosen, use the Deselect All command in the File menu, press CTRL-\ (backslash), or select a different file while Add does *not* appear in the status line.

Altering the Appearance of Displayed Files

While you can change which files are displayed in the files list by selecting the drive and directory, you can also change which files are listed and how they appear with File Display Options in the Options menu. When you select Options and File Display Options, DOS displays a dialog box that selects which files are listed in the selected drive and directory and how they appear. In the Name rectangle, you can use the DOS default of *.* or make another selection for the files to list. For example, you can type ***.BAT** to only display files with a .BAT extension. The Display System/Hidden Files option selects whether system files are hidden. This option is not initially selected so these files are hidden. The Descending Order option determines whether the files are listed in ascending or descending order according to the selection made with the Sort By option buttons. The default is not to select this option so the files are in ascending order. The Sort By options buttons select the order that the files are listed in the file list. You can select Name, Extension, Date, Size, or DiskOrder. The last one lists the files in the order they appear in the directory. The default is Name. When you select the OK command button, DOS updates the files listed to use these settings.

Selecting File Commands

After selecting the files you want to work with you can use the DOS Shell menu to perform commands on them. This lets you perform a DOS command without knowing its syntax since the DOS Shell will display dialog boxes for you to provide the information the command needs. Using the DOS Shell, you can copy, move, delete, view, rename, and change attributes of files. You can also open

files, associate files with a program, and search files for text. You can create and delete directories to organize the storage of your files and print a copy of a file that contains text with the File Print command. Many of the entries you will need to make in the dialog boxes for these commands are self-explanatory. After looking at the details for a few you will feel comfortable making almost any command selection from the menu.

■ ***NOTE:*** Remember, if you are in doubt you can always press F1 to get more information on the effect of a command before proceeding.

Copying Data

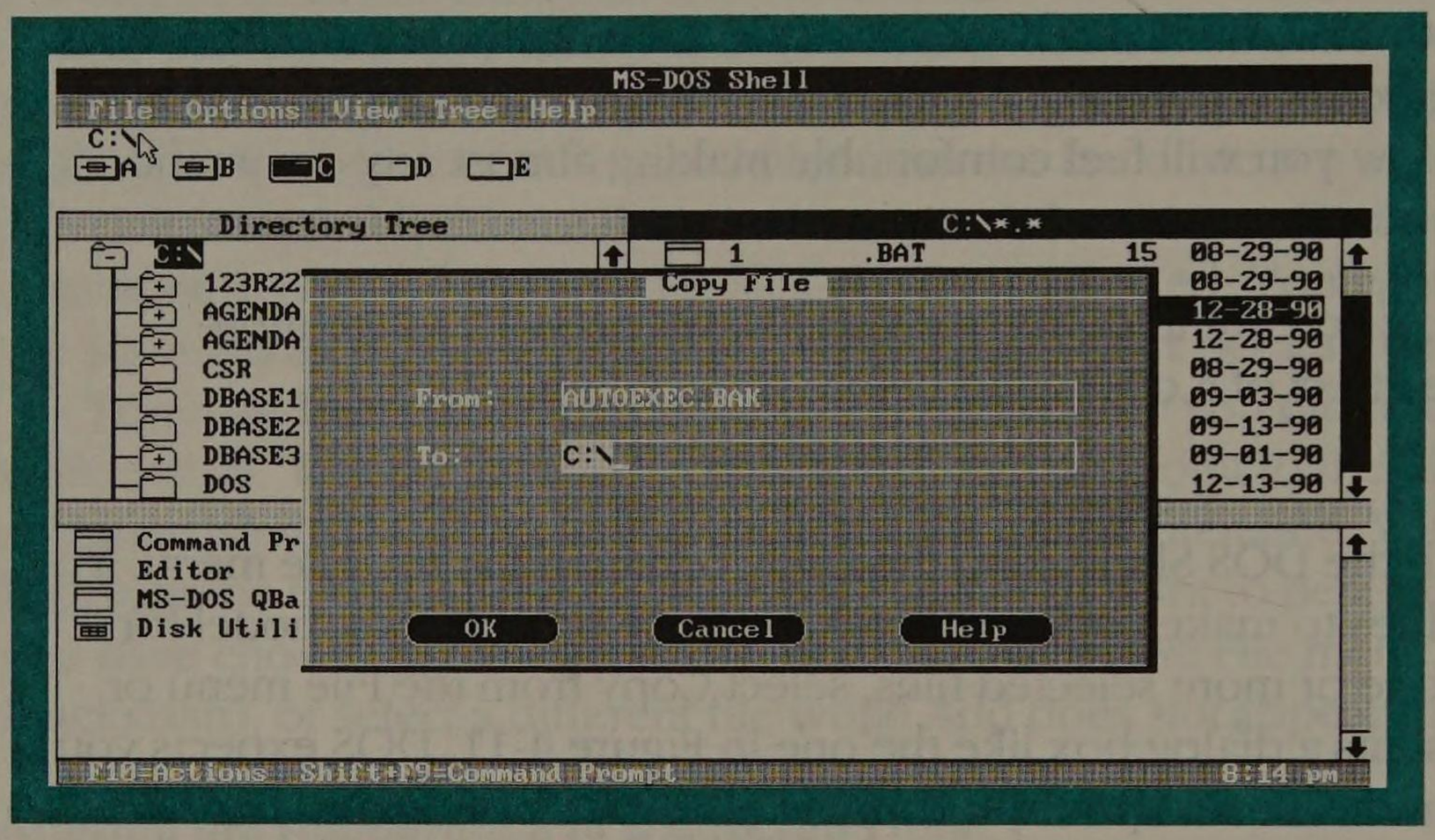
You copy files with the DOS Shell using the Copy command in the File menu. You will want to copy files to make backup copies or to use one file as a model for another. To copy one or more selected files, select Copy from the File menu or press F8. This presents a dialog box like the one in Figure 4-11. DOS expects you to make an entry in the To text prompt. You can press SHIFT-TAB or click the From text prompt if you want to delete one of the selected files listed to prevent it from being copied. In the To text box, you must supply the path information where the file will be copied. This is the drive letter, a colon, and then (possibly) a backslash and a directory name where you want the file copied. When you select the OK command button DOS will copy the file for you and update the file list and directory tree if necessary. In Chapter 7, you will learn how to use the DOS COPY command from the command prompt.

To copy files with a mouse, select the files and press CTRL while you drag the mouse to the drive and/or directory where you want the files moved. When you release the mouse button and CTRL, DOS displays a confirmation message. Tell DOS to copy the files by selecting Yes. Using a mouse to copy files requires that you make sure the drive and directory where you want the file copied is visible.

Moving Data

The DOS Shell allows you to move files by transferring files to another directory or disk. When files are copied to a new location they are deleted from the original location. To move one or more selected files, select Move from the File menu or press F7. This presents a Move File dialog box. DOS prompts you to enter in the To text prompt, the drive and directory where you want the files moved. You can press SHIFT-TAB or click the From text prompt and remove one of the files listed to prevent it from being moved. When you select the OK command button DOS will move the file for you and update the file list.

Figure 4-11: Dialog Box for the Copy Command from the File Menu



To move files with a mouse, select the files and press ALT while you drag the mouse to the drive and/or directory where you want the files moved. When you release the mouse button and ALT, DOS displays a confirmation message. Tell DOS to move the files by selecting Yes. Using a mouse to move files requires that the desired drive and directory is visible.

Deleting a File

When you finish using a file, you can delete it. You will want to do this when you no longer want the data in the file. To delete one or more selected files, select Delete from the File menu or press DEL. This presents a Delete File dialog box. You can press SHIFT-TAB or click the Delete text prompt and remove one of the files listed to prevent it from being removed. When you select the OK command button DOS will display confirmation dialog boxes to confirm that you want to delete each of the selected files. You can select Yes to delete the listed file or No to omit deleting the file. In Chapter 6, you will learn how you can use the DOS DELETE command from the command prompt.

Renaming a File

You can rename files in the DOS Shell so the filename meaningfully describes the file's contents. To rename one or more selected files, select Rename from the File menu. This presents a Rename File dialog box. For each selected file, DOS displays the current name and prompts you to enter a new filename in the New name text prompt. You must provide a valid filename, which has a maximum limit

of eight characters and an optional extension, which is a period followed by three characters. When you select the OK command button DOS renames the file for you and prompts you for the next new filename until all selected files are renamed.

Printing a File

Although the File menu offers a Print option, it will not work until PRINT has been used at least once during the DOS session from the DOS prompt. To use print for the first time you must exit the Shell and type **PRINT** at the DOS prompt. DOS will expect you to confirm PRN as the default print device or type another device name before pressing ENTER if you do not want to use PRN. You can return to the Shell by typing **DOSSHELL** then select a file and choose File Print to have it printed. Subsequent prints during the current DOS session only require that you select the file and choose File Print.

- ***NOTE:*** Attempting to print a file that does not contain text entries may cause your printer to behave erratically since some of the entries in the file may match printer control codes that cause your printer to eject paper or switch features.

Viewing a File

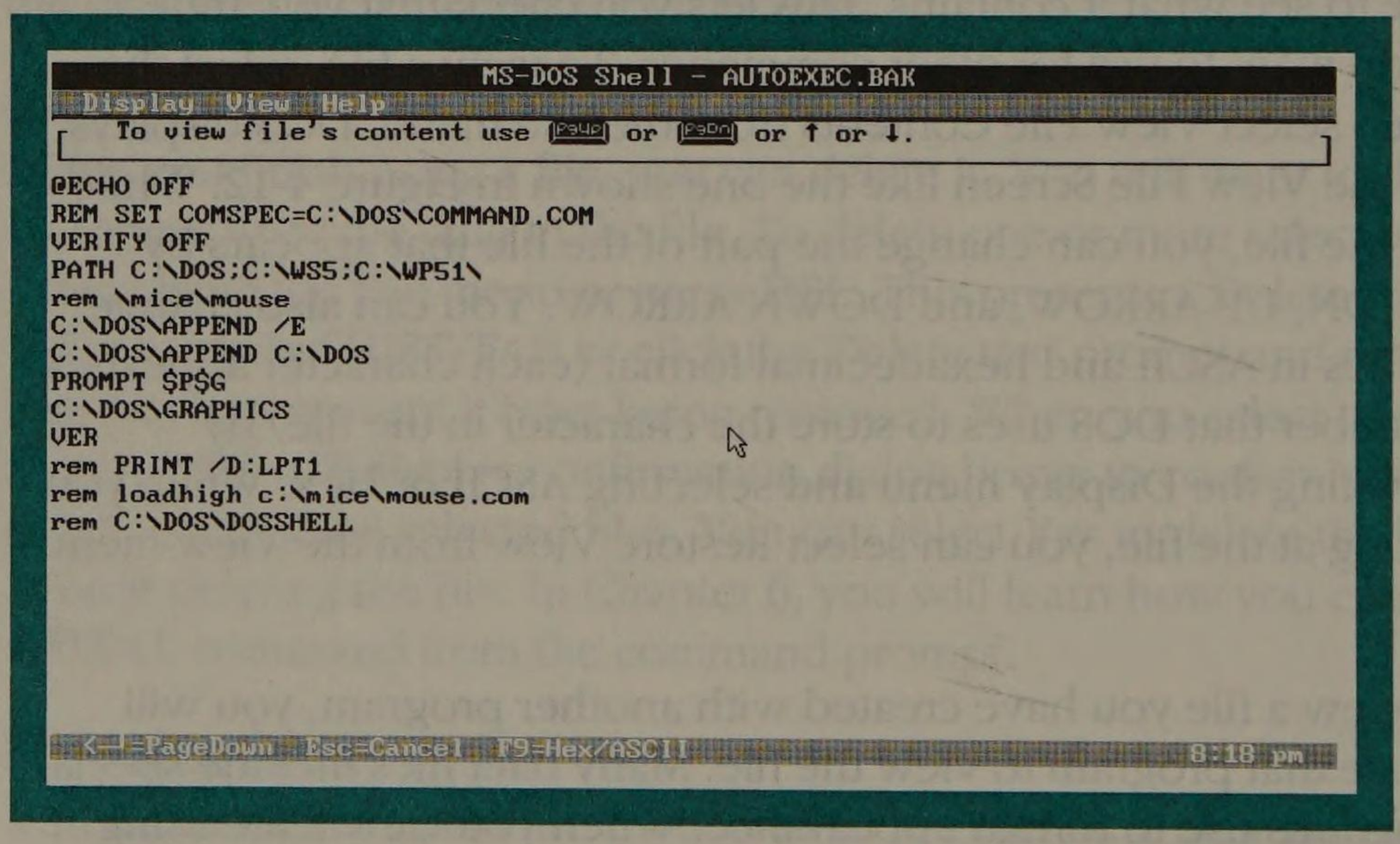
You can view a file to see what it contains. This lets you check that you are selecting the file you want to use for other commands. To view a file, select the file to view and then select View File Contents from the File menu. DOS displays the selected file in the View File Screen like the one shown in Figure 4-12. While you are looking at the file, you can change the part of the file that appears by pressing PGUP, PGDN, UP ARROW, and DOWN ARROW. You can also change between viewing files in ASCII and hexadecimal format (each character appears as its hexadecimal number that DOS uses to store the character in the file) by pressing F9 or activating the Display menu and selecting ASCII or Hex. When you have finished looking at the file, you can select Restore View from the View menu or press ESC.

- ***NOTE:*** When you view a file you have created with another program, you will probably want to use that program to view the file. Many data files include special characters that programs use to format appearance. When you view a file using the command described above, it will include the special characters rather than the formatting it represents. Also, most files that are part of a program are not designed to be read, and will seem unintelligible when you view them.

Changing File Attributes

A file can have four attributes. You can change them using the Shell. These attributes allow you to hide files; make them system files so DOS uses them to run the operating system; set the files as read-only so you cannot modify them; and archive the files, which lets you indicate specific files to be backed up. To change the attributes of the selected files, select Change Attributes from the File menu. If multiple files are selected, DOS presents a box that lets you select whether you want to set the attributes for the files individually or whether you want to set the same attributes for all files. If you are selecting the file attributes individually (the default), for each selected file DOS displays the filename and lets you select from Hidden, System, Archive, and Read only. The currently-selected attributes are marked with a triangle. You can add or remove an attribute by highlighting the attribute name and pressing the space bar or clicking on the attribute name. If you are selecting the file attributes as a group, DOS displays the four attributes and lets you add or remove them for the group by highlighting them and pressing the space bar or by clicking on the attribute. When you select the OK command button DOS changes the attribute for the one file and prompts you for the next one or changes the attributes for all of the files.

Figure 4-12: Viewing a File from the DOS Shell



You will want to reserve the file view option for looking at text files such as this batch file.

Running a Program

The programs in the bottom half of the window are programs you can use to provide other DOS features. These groups can include application software such

as 1-2-3 or Word for spreadsheet and word processing. You can run one of these programs in four ways. You can run a program by using the Run command in the File menu, run it directly from the task list, run it from the file list, or you can open a file with the extension it is associated with. This last option is covered in the next section.

To run a program with the Run command in the File menu, select the File menu and select Run. DOS displays a dialog box prompting you for the program to run. You must provide the path information and the filename to run. For example if you want to run WORD in the C:\WORD directory, you would type **C:\WORD\WORD** and press ENTER. This option for running a program is best used for programs that you do not usually run from the DOS Shell. This option is not good for programs listed in the task list.

When a program is listed in a task list, you can start the program by selecting it from the list. With a mouse, point to it and double-click on it. With a keyboard, press TAB until the task list is active, press the arrow keys to highlight the program you want to run, and press ENTER. Depending upon the properties assigned to the task, you may see a display prompting you for information. For example, when you select MS-DOS Editor, DOS displays a dialog box prompting you for the name of the file to edit. If the group containing the program in its task list is not the current group, you may need to change the group that is displayed in the task list before you can run the program. To switch between groups, double click on the group you want or use the UP ARROW and DOWN ARROW to highlight the group name and press ENTER.

A third method is selecting the executable file from the file list and running it. Executable files are files with a .BAT, .COM, or .EXE extensions. You can run one of these program by highlighting the program name in the file list and double clicking on it or pressing ENTER or selecting the Open command in the File menu. Using this option requires that you know which executable file you need to execute in order to run the program. As an example, the directory for 1-2-3 Release 3.1 has 13 executable files, but you will want to run the 123.EXE or LOTUS.EXE file to work with spreadsheet data.

While a program is running, it behaves exactly as if you did not use the DOS Shell to start it. You will not see the DOS Shell until you exit the program. For example, if you start 1-2-3 from the Shell, you can use it to perform all of the spreadsheet actions you want and when you type **/qy** to leave 1-2-3, you will return to the DOS Shell.

Associating an Extension with a Program

Most files you create are designed to be used with particular software. For example, if you have a 1-2-3 worksheet file, you will normally use it with 1-2-3. You can associate an extension with a program so that when you open a file with this extension, DOS loads the program and the data file. For example, if you open a Word file with a .DOC extension, DOS will immediately load Word for you and put the .DOC file in Word. DOS lets you associate up to 20 file extensions to a program although an extension can only be associated with one program. This lets you associate all files with a .WK1 extension with 1-2-3 Release 2, but you cannot also associate these files with 1-2-3 Release 3. To associate a file extension with a program, select a file or files in the file list that have the extension or extensions you want to associate and then select Associate from the File menu. For each file extension used by the selected files, DOS prompts for the program name to associate with these files. You must type the path information and program name. Another method of associating extensions with a program is by selecting the program and then selecting Associate from the File menu. DOS will prompt you for the extensions you want associated with the program. For example, you might type **DOC TXT PRT**. Notice that the entry does not include periods or asterisks.

Once a program is associated with an extension, whenever you open a file with the extension using the Open command in the File menu or press ENTER while highlighting the file, DOS runs the program and loads the selected file. If you later want to run a different program with files that are associated to another, you will need to start the program using another one of the methods described above. The DOS Shell will not let you open a file which is not associated with a program, so DOS will beep if you try opening the file by pressing ENTER twice or double clicking on the file in the file list.

To later remove an association between a file extension and a program, select one of the files with the extension that you want to remove from the association and select Associate from the File menu. When the Associate File dialog box appears, press DEL to remove the displayed associated program name and press ENTER. Another method is to select the program file from the file list and then select Associate from the File menu. When the Associate File dialog box appears, edit the file extensions to remove the association to the file extension you want to delete.

Searching

The DOS Shell has a command that makes it easy to find a file when you do not remember the directory where it is stored. When you use the Search command in the File menu, you can enter the description of the file you want to find in the

Search dialog box. This can include the question mark and asterisk wildcards. For example, you can use entries like MY_FILE, *.DOC, or ??????90.*. You can also select or deselect the check box to determine whether DOS searches all directories on the current disk or whether DOS only searches the current directory and any of its subdirectories. When you select the OK command button, DOS displays a list of all files that it finds that match the file specifications. Figure 4-13 shows the results of searching for executable files. This display remains on the screen until you press ESC to return to the prior display or use one of the commands in the View menu to change the view that appears on the screen.

Creating a Directory

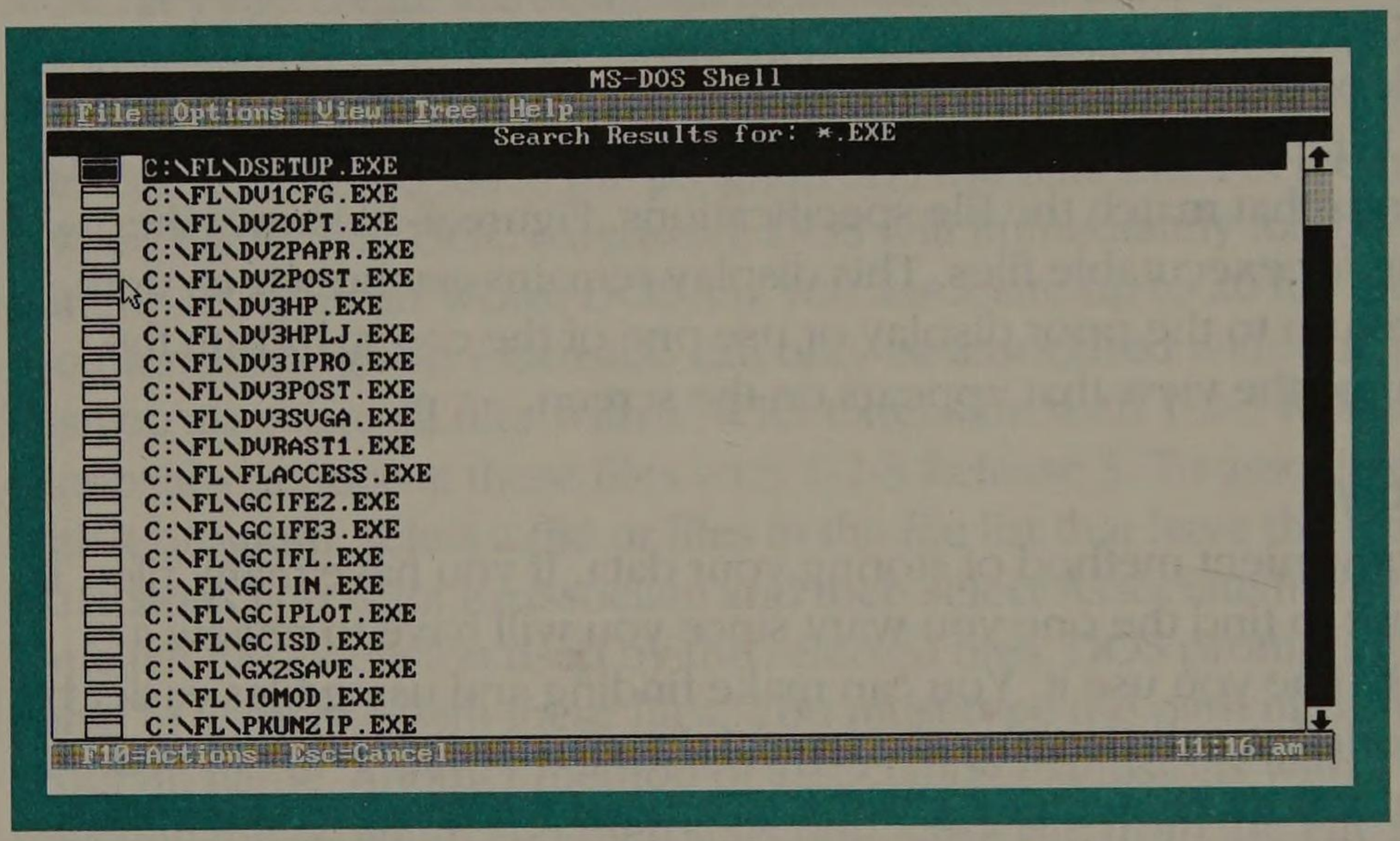
Disks provide a convenient method of storing your data. If you have many files, it will become difficult to find the one you want since you will have to review a directories list each time you use it. You can make finding and using files easier by creating subdirectories. A subdirectory is a logical division that divides the files stored on a disk into more meaningful groups. Chapter 3 went into details about subdirectories. If you are using a hard disk, you probably are using subdirectories so that the files each program uses are stored in different directories. Most software that is run on hard disks have their own installation programs that create hard disks for you. Creating a directory is as simple as using the Create Directory command from the File menu. Before you use this command, you must change the drive and directory displayed in the file lists so that the drive and directory you want to become the parent directory of the new one is listed. When you select Create Directory from the File menu, you must enter the name of the new directory. Directory names follow the same rules as filenames although most directories do not use an extension. When you select the OK command button, DOS creates the directory and updates the directory list if appropriate.

If you later realize you want to use a different name for the directory, you can use the Rename command in the File menu to change the name of the directory.

Deleting a Directory

Besides creating a directory, you can also delete one. You will want to delete directories after you have removed the files from them. DOS does not let you remove directories that still contain files or other directories. After you have deleted all the files and directories, you can delete a directory by selecting the directory to delete in the directory list and using the Delete from the File menu or pressing DEL. If the directory is not empty, DOS displays a message saying that access is denied and lets you try again or cancel the command.

Figure 4-13: Listing Executable Files Found with Search in the File Menu



Use a pattern to selectively view files like this.

Customizing the Shell

The DOS Shell has several settings that let you control how the DOS Shell behaves. You can change the screen mode to display the DOS Shell screen in text or graphics as well as change the number of lines. You can also change the colors the DOS Shell displays. You can select when DOS displays a confirmation message and whether DOS beeps, such as when you make an inappropriate selection.

Changing the Screen Mode

You can change the text or graphics mode from within the DOS Shell program or when you start the DOS Shell. To change the text or graphics mode from the Shell, select Options from the menu bar and Display from the Options menu. DOS will display the potential choices. You can select one of them and press ENTER or double click on one of the selections with the mouse. DOS updates the screen to use the new choice. When you have the selection you want, select the OK command button. Changing the screen mode this way lets you change the number of lines on the screen and whether the DOS Shell uses a text or graphics screen.

To change the text or graphics mode when you start the DOS Shell, include the /t or /g switch. The /t switch selects the text mode and the /g switch selects the graphics mode. After the t or g is a colon and one of these letters: l, m, h, v, s, or u. These letters represent different quality levels. You can remember these let-

ters as representing low, medium, high, very high, super high, and ultra high. As an example, to load the DOS Shell with the best graphics quality, you would enter:

```
DOSSHELL /G:U
```

When you select a text or graphics mode that is beyond the capability of your hardware, the DOS Shell uses the best quality available from the hardware.

Changing Colors

DOS has several color selections. You can either use one of these color selections the Shell provides or you can define your own. To use one of the preset color selections, select the Options menu and Colors. DOS lists the available color selections. Select the one you want from the list and choose the OK command button.

To create your own default color settings, you must edit the DOSSHELL.INI file as described in Chapter 10. You will want to copy the lines for one of the color selection choices which starts with selection= and modify the selection title and colors to use the colors you want. For each of the areas on the DOS Shell screen you can select the color of the text, the foreground, and the color behind the text, the background. The names of the different areas you can select and the areas they represent are listed in Table 4-2. You can use these colors: black, brightblack, white, brightwhite, blue, brightblue, red, brightred, cyan, brightcyan, green, brightgreen, magenta, bright magenta, brightyellow, and brown. You will want to become familiar with using the DOS Editor before you make changes. Also, copy the original DOSSHELL.INI file to another disk so you can return to using it if you make a mistake. After you modify the DOSSHELL.INI file, the next time you load the DOSSHELL program, you can select the color selection you have created.

Table 4-2: Names and Areas to Select Colors for the DOS Shell

Shell Area Name	Area It Represents
Base	File list, directory list, program list, and task list
Highlight	Highlighted items that are not chosen and Shell title
Selection	Items that are chosen
Alert	Warning messages
Menubar	Menu item highlighted

Shell Area Name	Area It Represents
Menu	Other selectable menu items, menu bar and status line
Disabled	Menu items you cannot select
Accelerator	Keys you can press in place of selecting menu items
Dialog	Dialog boxes
Button	Command buttons in dialog boxes
Elevator	Box on scroll bar
Titlebar	Top bar with the Shell name
Scrollbar	Scroll bar
Borders	Borders of dialog boxes
Drivebox	Disk drive area
Driveicon	Disk drive icons

Silencing the Shell Beep

The DOS Shell is set to "beep" whenever you make an error or something happens where DOS wants your attention. You can silence it by starting the DOS Shell program and including the /n switch as in:

```
DOSSHELL /N
```

You can also use the /n switch to use the bell at a later point. Since this is the default you normally do not need to use this switch.

Displaying or Hiding Confirmation Messages

DOS will normally display confirmation dialog boxes when you delete or replace files or directories. DOS also displays confirmation boxes when you move or copy files with a mouse. You can prevent DOS from displaying these messages by selecting Confirmation from the Options menu. The Confirmation dialog box has three options which select when DOS displays a confirmation message. By deselecting one of the options, you prevent DOS from displaying a warning message when the event on the right occurs. The three confirmation options are Confirm on Delete, Confirm on Replace, and Confirm on Mouse Operation. The first displays a confirmation message before deleting files; the second displays a confirmation message before overwriting an existing file; and the third option displays a confirmation message before copying, dragging, or moving files with a

mouse. When you press ENTER or click on the OK command button, the DOS Shell uses these new confirmation settings.

At a Glance

The DOS Shell lets you perform many DOS commands without knowing the commands. You can select the files you want to use, the actions you want to take with them, and the programs you want to run directly from the screen. The DOS Shell provides an easier-to-use interface than the DOS command prompt. Some of the information you have learned in this chapter includes:

- **Move from area to area in the Shell—TAB**
- **Get Help on any topic—F1 and Help menu**
- **Change the drive for the file list—Click with the mouse or press TAB and an ARROW key followed by ENTER**
- **Change the directory—Click with the mouse or press TAB and an ARROW key followed by ENTER**
- **Expand and contract the directory tree—Tree commands**
- **Selecting multiple files—SHIFT and CTRL**
- **Run a program—Select Run from the File menu, or select the program from the file list or from the task list**
- **Copy, move, rename, delete, view and change file attributes—ALT-F and appropriate command**
- **Associate a file extension with a program—ALT-F then Associate**
- **Search for files—ALT-F Search**
- **Create and delete directories—Create directory and Delete in the File menu**
- **Change the screen display, colors, and confirmation settings—ALT-O plus appropriate selection**
- **Change beeps, mouse buttons and displays with switches—/b, /n, /s, /t:, and /g:**

Chapter 5

Using the Shell to Work with Programs and Utilities

Programs are a set of stored instructions that perform tasks. Some programs are part of DOS and offer features such as the ability to edit or print a text file or copy a disk full of data files. Other programs are purchased separately and offer support for a variety of different applications. Some applications you might use are: word processing to create memos and reports, spreadsheets to compute budgets and sales projections, or graphics to create presentation quality materials. With the DOS 5.0 Shell you have the ability to run two or more of these programs at the same time. This allows you to work concurrently on multiple tasks and switch between various applications. Each of your applications will remain active in memory. You will control the application you want to have actively processing at any point in time, but you can switch to any of the other active applications with just a keystroke or two. The Shell supports this multitasking by allowing you to add programs to the active task list and provides the support for controlling the active application.

The Shell also provides access to a set of disk utilities. Like the Shell file management features you discovered in Chapter 4, the disk utilities are usable without the need for memorizing the syntax of DOS commands. You will find that they offer you a quick way to perform repetitive disk tasks.

What You'll Learn

In this chapter you will learn about programs that can be executed from the DOS Shell. You will learn to use the Switcher to move between the different programs and even how to create your own custom program entries and program groups. You will also learn about how to use the DOS utilities. Some of the specific tasks you will master in this chapter are:

- **Start a program**
- **Use the Switcher to run two or more programs at once**

- **Open or close a program group**
- **Add a program to the main program list or a group**
- **Add a program group to the main program list**
- **Customize program properties**
- **Add a password to a program or group**
- **Associate a file type with a program**
- **Run the DOS disk utilities**

Working with Programs

Without programs your computer hardware would be useless because programs provide the instructions that enable your computer to handle all types of tasks. The Shell makes it easy to start one or more programs and to switch between them. The Shell's ability to maintain information on multiple active tasks allows you to activate several programs and switch between them quickly. Although only a few programs are available in the initial program list in the DOS Shell, you can add your own programs and groups of programs to this list to make these features more valuable to you.

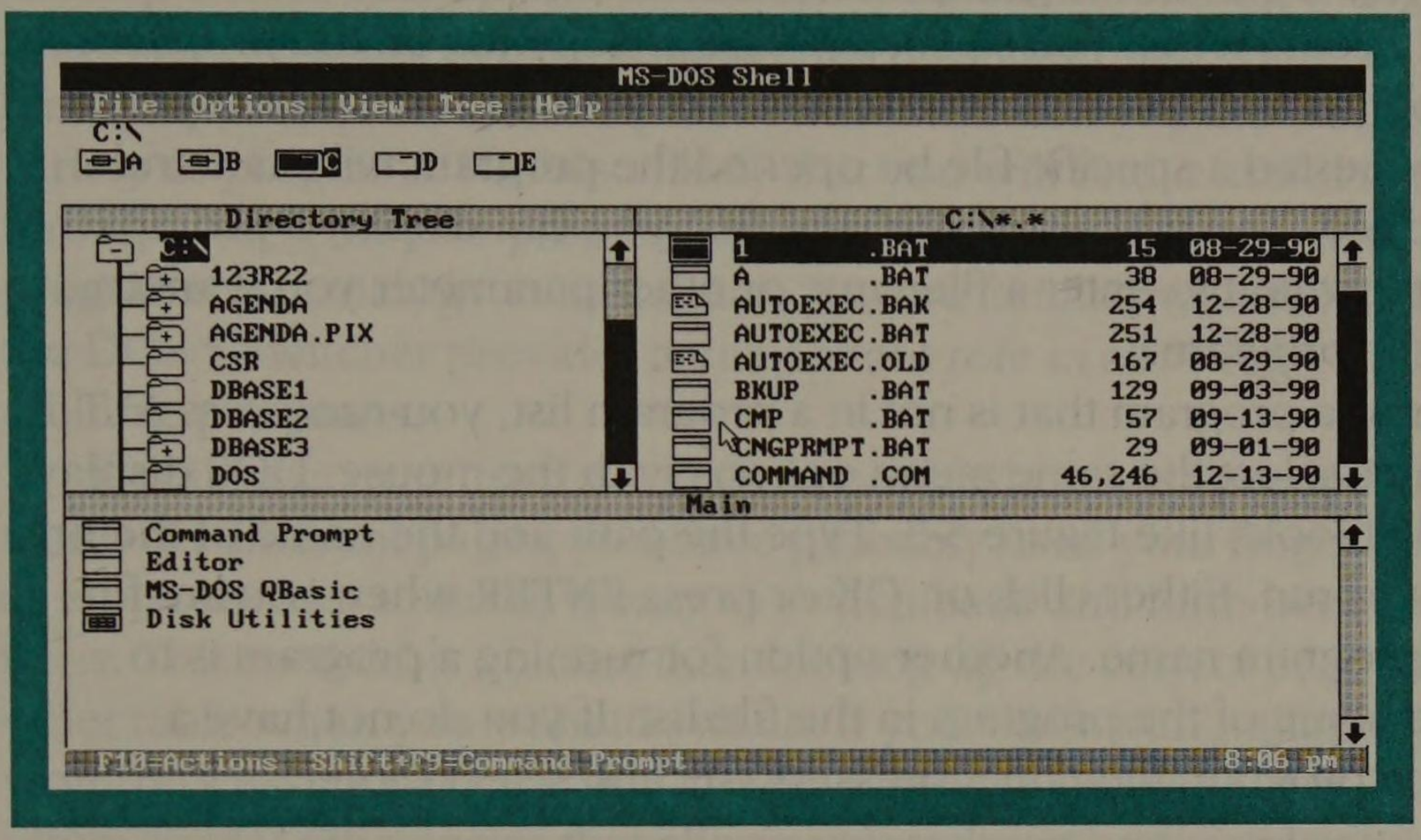
Understanding the Program Manager Components and Capabilities

Initially the Program Manager only lists a few options. You can use it to activate the DOS prompt, to use the System Editor or QBASIC, or use the group of programs called the Disk Utilities.

The icons in front of each option in the program list indicate whether the option is a program or a program group. An icon that looks like a rectangle with a horizontal line across the top is an individual program. An icon of the same shape with six boxes in the lower part indicates a program group. A program group can contain other groups or individual program entries. Once a group is selected, the groups and items within the selected group will display. The first three icons in Figure 5-1 are program icons and the last icon is a group icon. If you are using a text based screen, these icons will not appear.

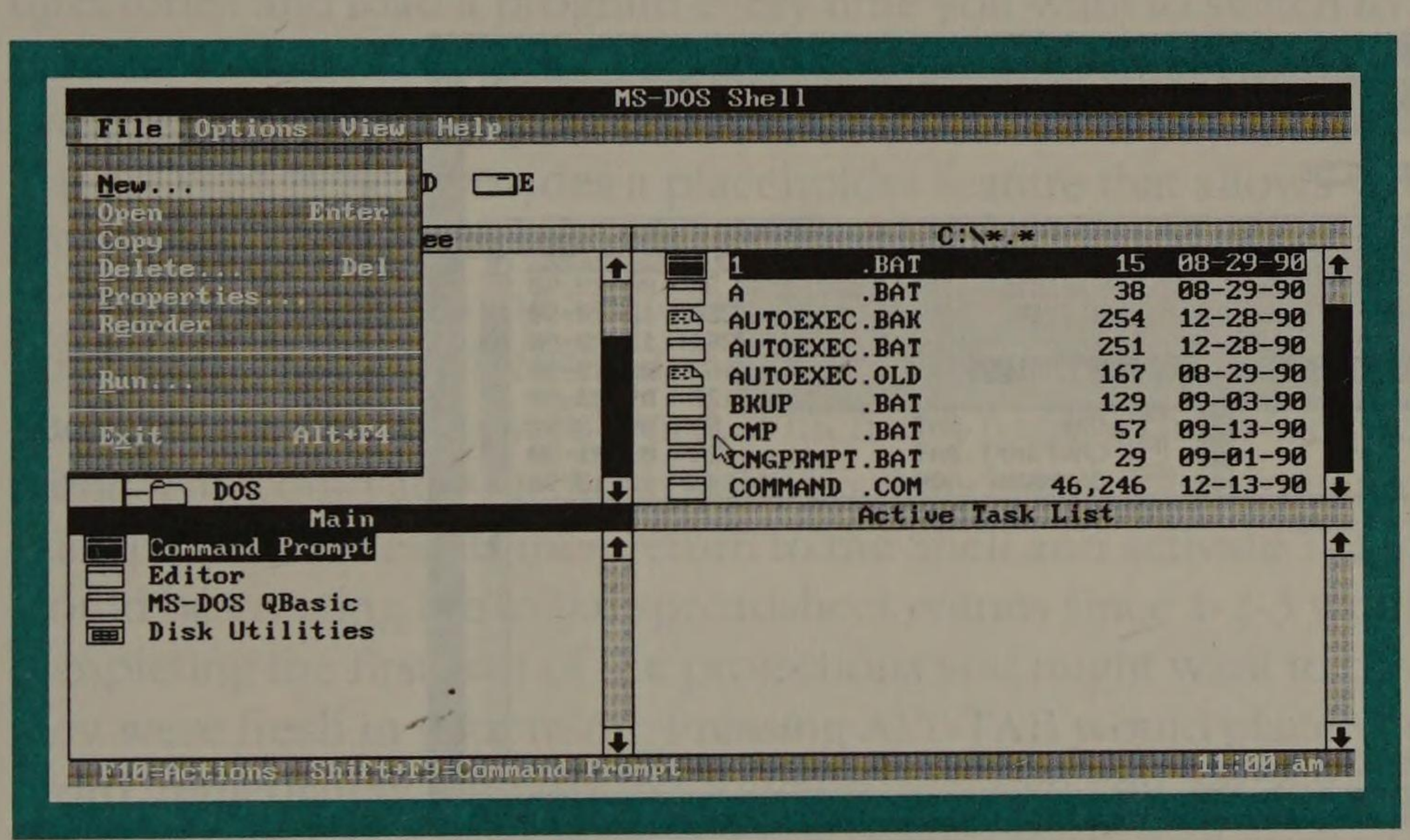
The menu options available when the program list is active are more limited than the ones presented when the file list is active. The Menu Bar options include only File, Options, View, and Help. The selections available under the File option are also more limited. Each of the File options presented in the menu shown in Figure 5-2 pertain to tasks performed with programs.

Figure 5-1: Main Program List Before Additions



The four options shown are available for everyone when DOS is installed.

Figure 5-2: Program File Menu Options



Note the more limited options in the file menu when the program list is active.

Running Programs

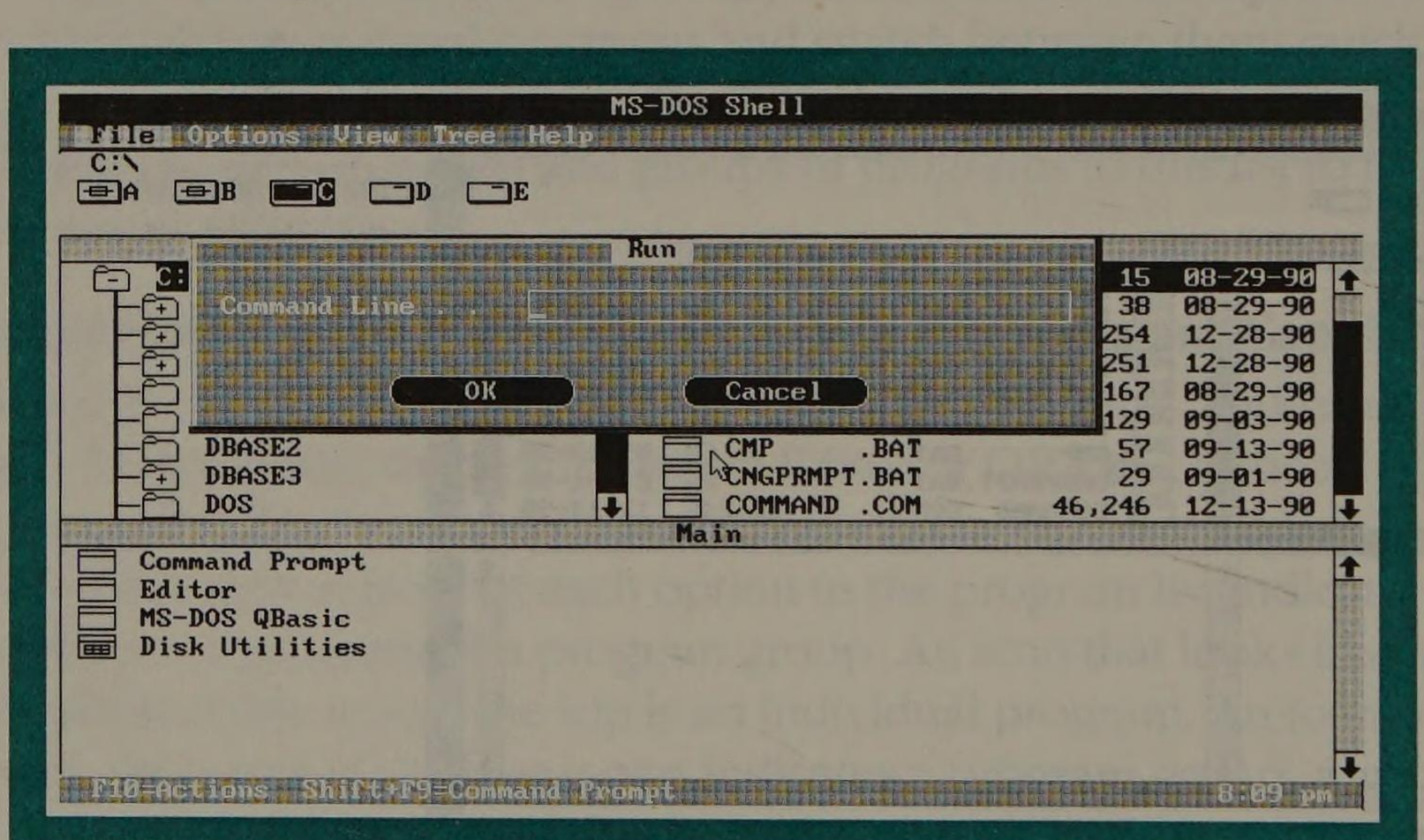
When you select a program with a double click or by highlighting and pressing ENTER, DOS starts the program. You can also highlight the program with TAB and the ARROW keys or a single click of the mouse and press ALT-F and type **O** to open the program file. When you perform the same action with a program group,

DOS displays a list of the groups or programs within the group you selected. You can then select a program from the group that you want to run.

The action that occurs when you select a program depends on the properties assigned to the program. It is possible that it will start with no further interaction from you. If you requested a specific file be opened the program will start and open the requested file. You will also learn later that you can display a prompt message that will allow you to enter a filename or other parameter you want to use when starting the program.

If you want to run a program that is not in a program list, you can press ALT-F and type **R** for Run or select the same menu options with the mouse. DOS displays a Run dialog box that looks like Figure 5-3. Type the path and the name of the program that you want to run. Either click on OK or press ENTER when you are finished entering the program name. Another option for running a program is to double click on the name of the program in the file list. If you do not have a mouse, highlight the program filename, press ALT-F, and select Open.

Figure 5-3: Run Dialog Box



Command is entered in the command line text box just as you would enter it at the DOS prompt.

You can also apply password protection to the program. This will prevent unauthorized users from running the program.

When you exit a program you started from the DOS Shell, you will be returned to the Shell when you exit the program.

Understanding the Capabilities of the Switcher

In the previous section, the discussion focused on running a single program. Although this is the normal situation for new computer users, as you become more dependent on the computer for information you are likely to work with a variety of programs. It is also likely that you will find it useful to switch back and forth between the two programs as you reference data provided by both programs. An analogy to a situation you are familiar with will help explain how the DOS 5 Switcher provides an important role in allowing you to work with multiple programs.

If you have ever used a reference book where you needed to refer to two tables on different pages, you have probably used your finger or a piece of paper as a placeholder to make it easy to switch back and forth between the two locations. This method is quicker than looking up the correct page location for the other table when you want to switch. If you work with multiple programs such as word processing package and a spreadsheet or database, the DOS Shell Switcher offers similar efficiencies that allows you to keep both application programs active and switch between them. This means you do not need to take the time to change directories and load a program every time you want to switch to another program. Part of memory is allocated to each application and although only one is actively processing data at any point in time, both are active and available for your use. The DOS Switcher provides a placeholder feature that allows you instant access to any active program and its data.

In practical terms the Switcher makes it easy for you to create a sales projections with a package like the 1-2-3 spreadsheet, and to describe the results with a package such as Microsoft Word, which can handle the word processing document. You could activate Microsoft Word and start a document for describing the sales projection results then return to the Shell and activate 1-2-3. The next step would be creating the initial spreadsheet entries since 1-2-3 would be active. After completing the first part of the projections you might want to describe them while they were fresh in your mind. Pressing ALT-TAB would place you in Microsoft Word immediately after the introductory material you had entered previously. When you finished describing the initial projections and wanted to continue with the 1-2-3 model, pressing ALT-TAB places you in the sales projection worksheet with the cell pointer in the same cell where it was located the last time you used the worksheet.

If you need more than two applications and have sufficient memory you can have additional applications as active tasks within your system. Every time you press ALT-ESC, with the DOS Switcher active, DOS will switch you to the next active application. You can press SHIFT-ALT-ESC to go to the previous application.

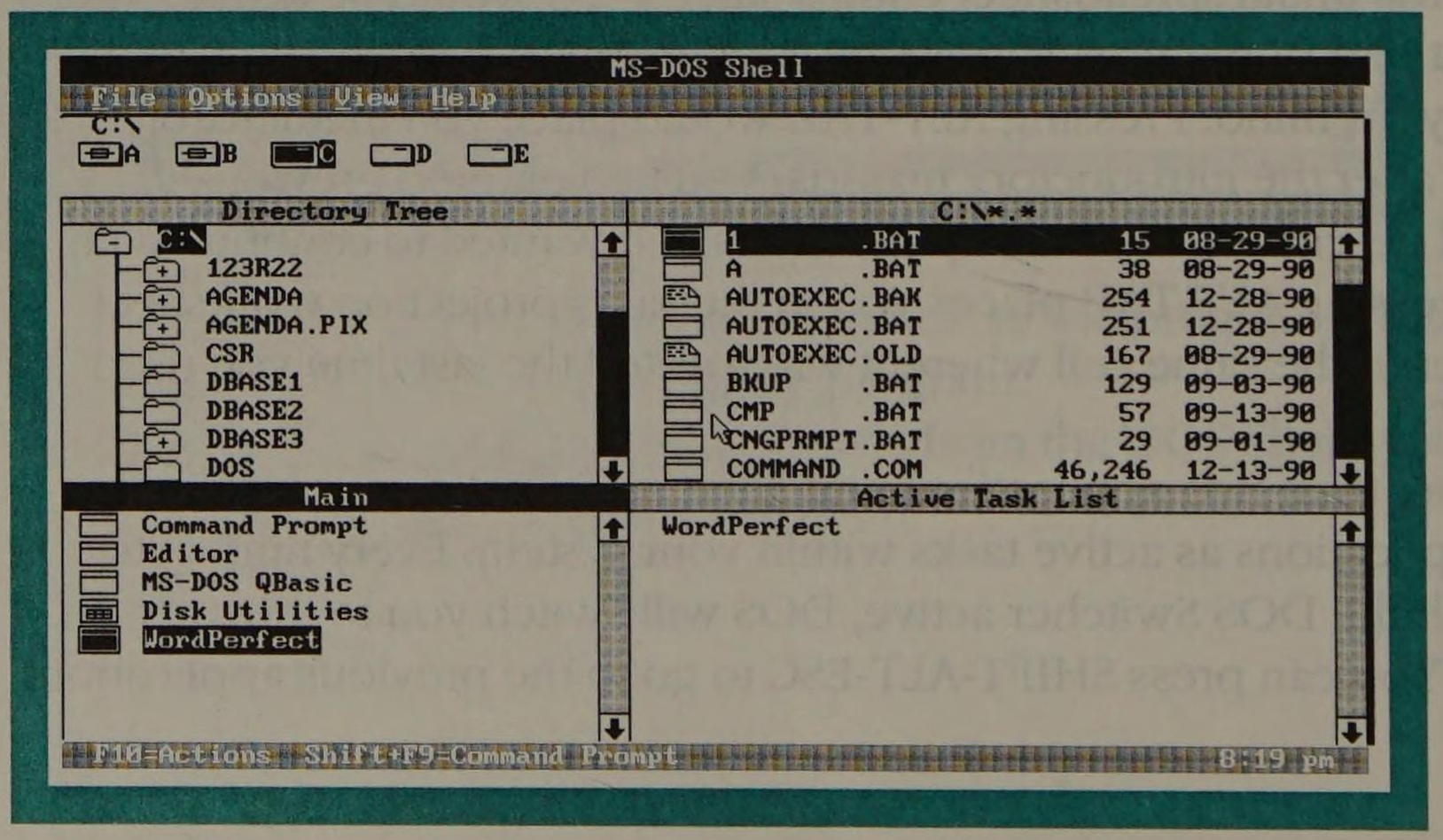
Activating the Switcher With the Switcher active you can have multiple programs in the active tasks list. This means each program occupies a portion of memory and can become the current application without a need to change directories or load the software.

There are two ways you can tell if the Switcher is enabled. First, the bottom of the screen will display the active task list on the right side if the default view of Program/File Lists is in effect. If only programs are displayed, the list is on the right but is not restricted to the bottom of the screen. If only file lists are displayed, the active task list does not appear on the screen. The second indication is always available regardless of the View option chosen. When you activate the Options menu with ALT-O, the Enable Switcher option has a diamond on the black dot to the left of the option when it is already enabled. Selecting it again at this point would disable the Switcher.

If the Switcher has not been enabled you can click on Options in the menu bar, then click on Enable Switcher if you are using a mouse. From the keyboard you would press ALT-O and type **E** to enable it.

Starting an Application You can start an application program to use with the Switcher once you have enabled the Switcher. All of the options described earlier are also options for starting a program that you plan to use with Switcher. This means you can double-click on an option with the mouse, highlight the selection and press ENTER, or use the RUN command. Using SHIFT-ENTER will add the program to the active task list yet remain in the Shell. When you return to the Shell you will see each of the active tasks shown in the active task list, just as you see WordPerfect in Figure 5-4.

Figure 5-4: Active Task List



Note the program item for WordPerfect has been added.

Switching Between Applications You can switch between active tasks as often as you like as long as you do not exit from the tasks. You will want to use the special key sequences described in Table 5-1 to make the switch between the programs and the Shell.

Table 5-1: Key Sequences to Use with Switcher

Key Sequence	Effect
ALT-ESC	Goes to the next application
ALT-TAB	Toggles between two tasks
CTRL-ESC	Returns to the DOS Shell
SHIFT-ENTER	Start a program and add it to the task list without leaving the Shell

Deleting Applications When you no longer need an application you will want to remove it from the active task list. This allows you to use the memory for other applications. You do not need any special Shell command to remove the application. All that is needed is to end the application. The technique used will vary based on the application in use. For some applications, a function key will handle the task while others require a menu selection and confirmation. Table 5-2 shows a few popular application programs and the technique used to end them.

Table 5-2: Exiting from Popular Application Software

Program	Method of Exiting
1-2-3	Quit Yes
dBASE IV	Exit
Paradox	Exit Yes
Quattro Pro	File Exit or CTRL-X
Word	ESC QUIT
WordPerfect	F7
WordStar	File Exit

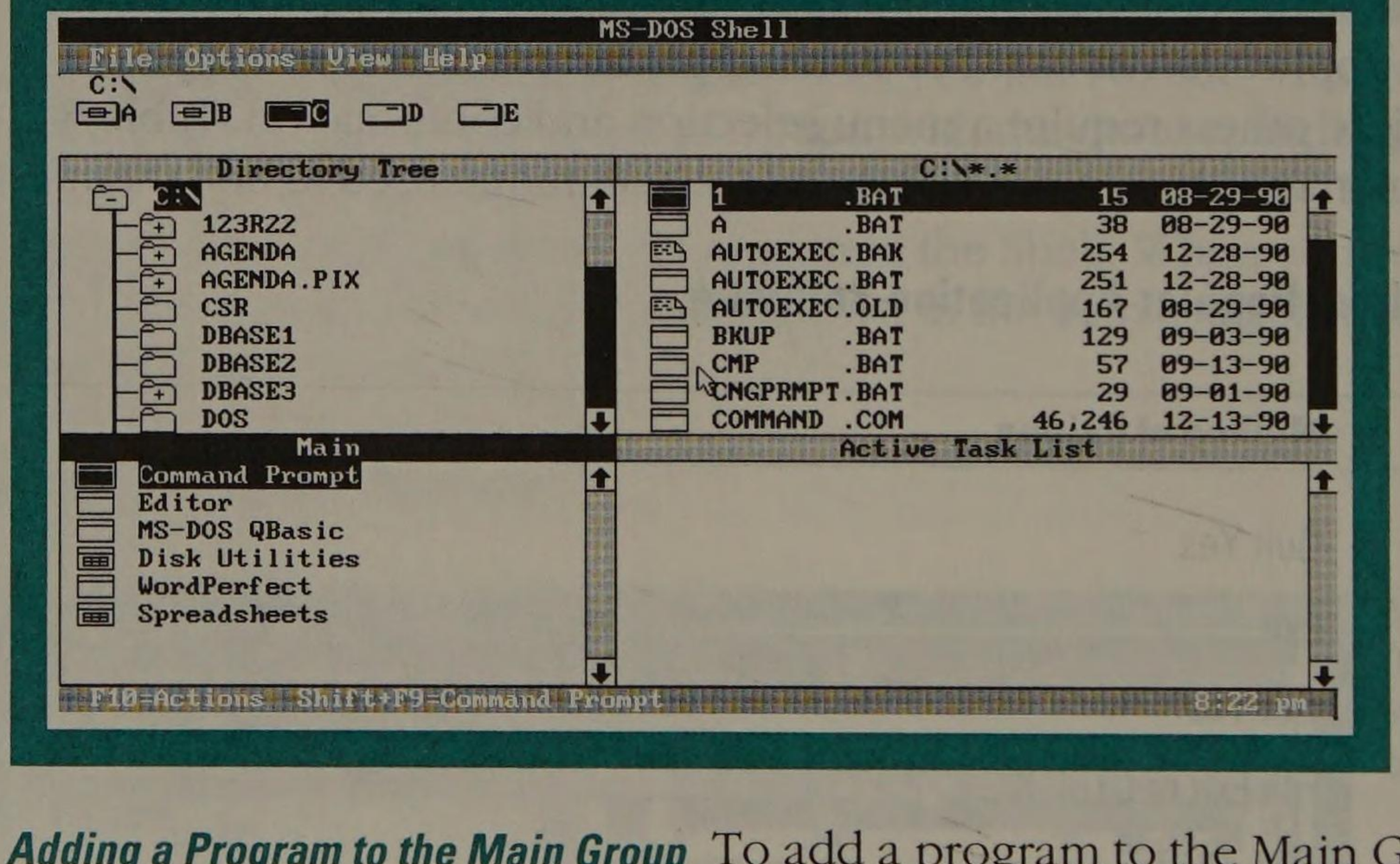
Although most applications have sufficient error handling capabilities so they will not end abruptly, you may occasionally encounter an error condition such as a bad or missing disk that the application cannot support. The first step is to return

to the Shell with CTRL-ESC. Next, delete the failed application from the Active Task list by highlighting it with a mouse click or the ARROW keys and selecting Delete from the File menu. It is also a good idea to activate each of the other applications and exit each one in an orderly fashion then reboot DOS to insure no problems exist in memory.

Setting Up Your Own Programs and Program Groups

To make the Program Manager and Switcher more useful you will want to customize them with the applications you use most often. You can add individual programs that you use frequently as options in the Main group, options that display all the time. If you have a group of related programs you might want to create an entry for the group in the Main program list. The procedure you use is similar for both situations. Looking at the steps needed to add WordPerfect as a program and Spreadsheets as a group will clarify the process. The Main group entries will look like Figure 5-5 when the additions have been completed.

Figure 5-5: Main Program Group with a New Program Item and a New Program Group



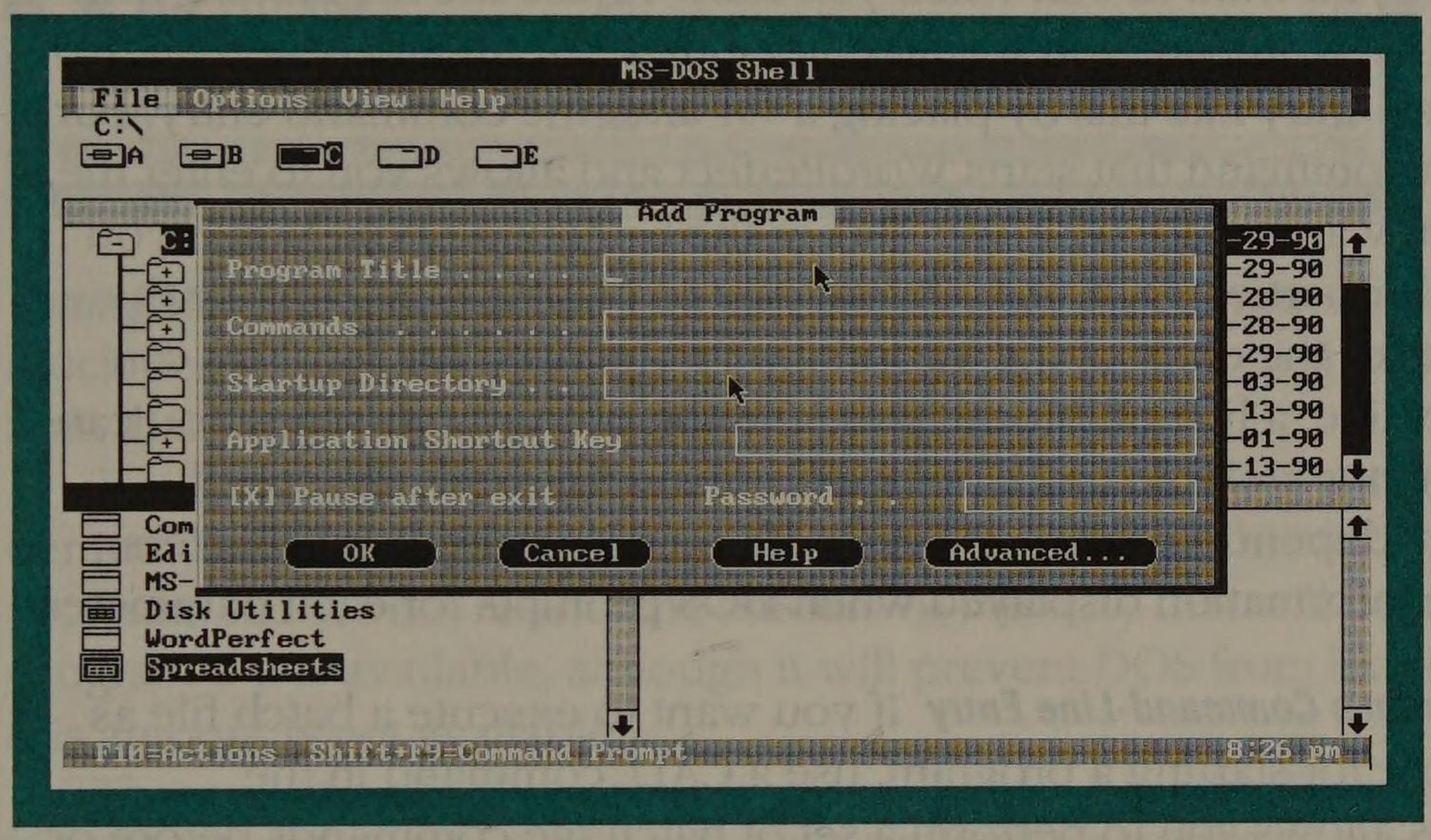
The icon for a program group is different than the one for a program item.

Adding a Program to the Main Group To add a program to the Main Group select File New from the menu. The New Program Object dialog box allows you to add either a Program Group or a Program Item. The selection with the black dot will be the one chosen when you press ENTER or click on OK. Use the mouse or ARROW keys to change it if necessary. If you choose Program Item, the Add Program dialog box like the one shown in Figure 5-6 will appear. This box is used for entering basic program properties such as the program name, its startup

directory, and the required command. This is the same box that will appear later if you decide to change the properties for the program. Table 5-3 describes each of the basic entries you can make in this dialog box. The next section discusses the use of parameters within command entries.

The advanced property options you specify for a new program allow you to add help text, indicate memory requirements, specify the video mode, and indicate special key uses for the program. You can choose these by selecting the Advanced option button at the bottom of the Add Program dialog box. These options will be discussed in detail after the information on special command line entries.

Figure 5-6: Add Program Dialog Box



You must supply the Program title and Command.

Table 5-3: Program Properties

Program Property	Meaning
Program Title	A title that describes the program that can be up to 74 characters long.
Commands	One or more DOS commands that you want executed to start the program. Multiple commands are separated by semicolons (;). The maximum length of the entry is 500 characters. Parameters such as %1 can be used in this entry.
Startup Directory	The directory that must be active to start the program. DOS changes to this directory before starting the program.

Program Property	Meaning
Application Shortcut Key	Key combination with ALT, CTRL, or SHIFT for starting the program.
Pause after exit	Prompts you to press a key to return to the Shell when you exit.
Password	An entry as long as 256 characters with capitalization of the entry part of the password.

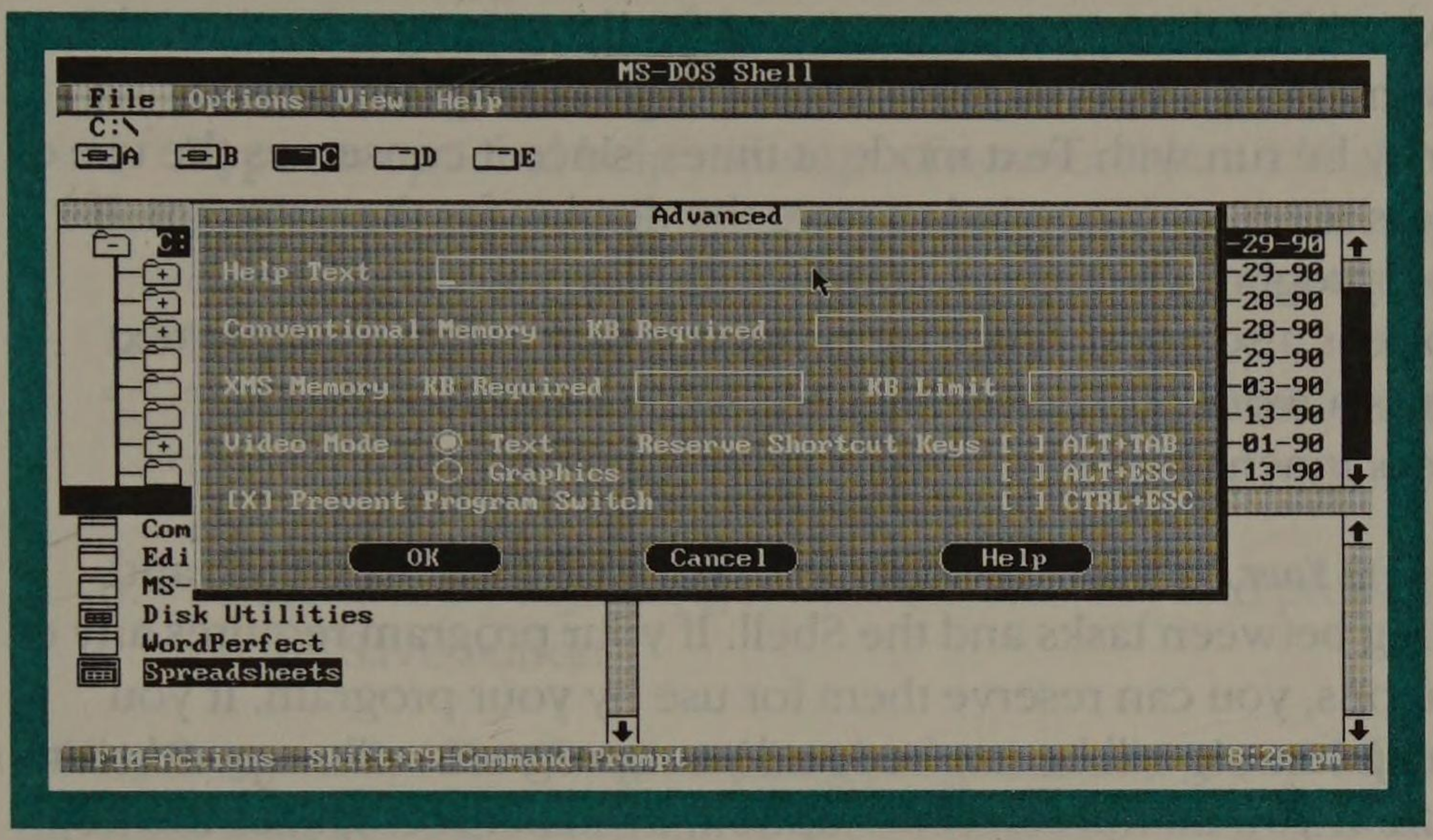
Using Parameters in the Command Entry for a Program You can use the replaceable parameters % 0 through % 9 if the program you are adding accepts parameter entries. For instance most word processing packages allow you to specify the name of the file that you want to edit when you start. To use the file MEMO with either WordPerfect or WordStar, you would enter either WP MEMO or WS MEMO. You can allow for an entry like this by placing a %1 after the command entry. For example, to enter a command that starts WordPerfect and allows you to enter the name of the file you want to use type **WP %1** in the Command box. When you finish entering the Properties, DOS asks if you want to display a prompt for the user when it is time for the entry of the information to be used for %1. You can supply a dialog box title, a line of information, and the prompt message. You can use the same parameter multiple times in the command line and also specify multiple parameters depending on your needs. If you use multiple parameters, you can supply the information displayed when DOS prompts for each parameter.

Running Batch Files from a Command-Line Entry If you want to execute a batch file as part of the procedure for starting a program, use a CALL command in the command line. This allows you to perform a set of batch file commands before or after executing your program. The command entry, **WP; CALL BKUP** executes WordPerfect and then executes BKUP. The command entry **CALL PREPARE; WP** runs PREPARE before running WordPerfect.

Adding Help Text for a New Program If you want help text to display when you highlight a program and press F1, you can add it through the first entry in the Advanced dialog box shown in Figure 5-7. You can enter as many as 256 characters. DOS will split your entry to fit with the help dialog box that will display. You can also control line breaks yourself by inserting an ampersand anytime you want to start a new line. An entry of **Starts WordPerfect&with my to-do list open** will display as:

Starts WordPerfect
with my to-do list open

Figure 5-7: Advanced Properties Options



This screen is optional and will not be completed for may of your program items.

Setting Memory Requirements You can use the Advanced options to specify how much conventional and extended memory must be free to start a program and the limit that can be allocated to the program.

The standard setting for conventional memory requirements is 384K. This number indicates the amount of memory that must be available for DOS to start the program. It does not restrict the amount of memory that will be allocated to the program if it is available, although it will prevent DOS from loading a program if this amount is not available.

There are two XMS memory options you can specify. They control the allocation of extended memory to a program that follows LIMS (Lotus Intel Microsoft Specifications) for extended memory use. The first setting, KB Required, is used to indicate the amount of extended memory the program must have. For most programs this number is 0 since even those programs that can use extended memory seldom require it. If you supply a number other than 0 and the amount specified is not available, a message will display indicating that it is not possible to start the program. The second option, KB limit, allows you to restrict the amount of extended memory allocated to a program. Since some programs will allocate all of the available memory to themselves whether the program will need it or not this allows you to save some extended memory for use by other tasks. Setting the limit to 0 prevents a program from using extended memory even if it is capable of doing so. A setting of -1 allows you to make available the maximum amount of extended memory that is currently free without knowing how much is available.

Specifying the Video Mode You can choose between Text and Graphics mode to control the use of the display by the program. Some programs only use text displays and Text would be the appropriate choice for these programs since this option requires less memory than Graphics. Even programs that support a Graphics display may be run with Text mode at times, since it conserves the use of memory because less memory is needed to store the display for the program. If you want to utilize a program that uses a graphics display you can choose Graphics. This choice is also appropriate for programs that utilize multiple text display pages such as a word processor that allows you to switch displays between different documents.

Using Reserved Keys with Your Program Switcher reserves ALT-TAB, ALT-ESC, and CTRL-ESC for moving between tasks and the Shell. If your program requires any of these key combinations, you can reserve them for use by your program. If you select one of these options, it will be marked and your program will respond to it in its normal fashion. If you do not select the option, Switcher recognizes the key combination and takes its normal action with no effect on the program.

Disabling the Switcher for a Program You can disable a programs's ability to switch to the Shell or another program. When you select the Prevent Program Switch, you will need to exit from the program in order to return to the Shell. From the Shell you can select the next task you want to make current.

Changing Properties for Existing Programs You can change program properties that you entered when you first added a program or you can add new properties with the File Properties command. After highlighting the program in the program list that you want to change, select File Properties. The box that originally appeared as the Add Program dialog box appears again, but now the name is Program Item Properties. You can change any of the entries in this box or choose Advanced to see the Advanced dialog box. Choosing OK when you are finished ensures that your changes will affect the next use of the program. The only group where you cannot change properties is the Main program group.

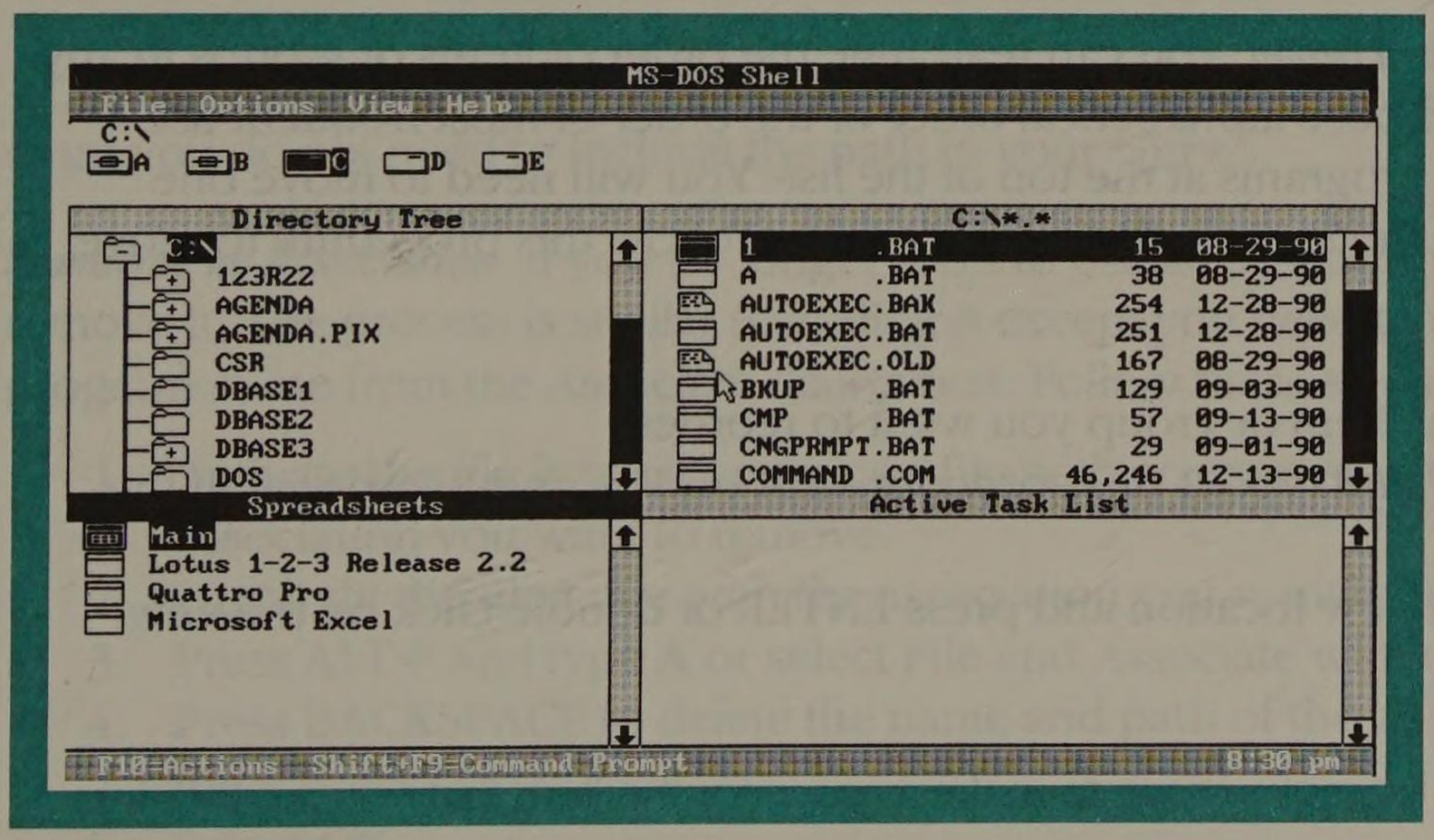
Adding a Program Group You can add a group to the Main program group or any other group. The group that you want to add the group to must be displayed in the program list and the program list section of the screen must be active. To add the group follow these steps:

1. Press ALT-F and select New or use the mouse to select File and New. (The New Program Object dialog box displays on your screen.)

2. Use the ARROW keys to select Program Group or click Program Group with the mouse.
3. Type a title of no more than 74 characters for the Group and press TAB or click the help message.
4. Type an optional help message of no more than 256 characters. (This message is entered in the same fashion as a program help message. The & character can be used to make the text which follows display on the next line.)
5. Type an optional password of 255 characters or less, noting your use of upper- and lowercase letters since capitalization is part of the password entry.
6. Select Save by activating it with the TAB key and pressing enter or clicking on the Save button.

Adding Programs to a New Group Adding programs to a group other than Main is no different than adding a program to the Main group. After activating the group that you want to add the program to, follow the steps that provide step-by-step directions for adding the program. Figure 5-8 shows the programs added to the Spreadsheet group. The customization options available for a program in the Main group are also available as options for programs in other groups. In addition to adding programs to the Main group or your own groups, you can also make additions to the Disk Utilities group.

Figure 5-8: Spreadsheet Programs



Three different program items added to the spreadsheet group.

Copying Programs to Other Groups As your customization of the Shell progresses, you might want to set up new groups and copy entries from an existing group to a new group. Follow these steps to complete the copy operation:

1. Display the group that currently contains the program that you want to copy and highlight the program.
2. Press ALT-F and type **C** for Copy or make the same selections with the mouse.
3. Supply the password if prompted.
4. Activate the group that you want to copy to.
5. Press F2 to complete the copy operation.

Deleting Programs and Groups To delete a program from a group, highlight the program and press ALT-F and type a D for delete or use the mouse to make the same selections. You will need to confirm that you wish to delete the entry from the program list by selecting option 1. Although the program itself still exists, there is no longer a menu selection to start it. Any properties defined for the program will be deleted when you delete the entry. If the program is password protected you must supply the password before deleting any information.

You can use the same procedure for deleting a group. When you delete the group the entry for each program item in the group disappears. This includes the properties defined for each of these items.

■ ***NOTE:*** A shortcut that works for deletion is to press the DELETE key. Although you still need to confirm your decision, it is quicker than a menu selection.

Reordering the Items in a Group You can rearrange the order of items or groups in any group. You might choose alphabetical order or the order of most frequent use with the most used programs at the top of the list. You will need to move one entry at a time until the list is in the desired order. Follow this procedure to move each item:

1. Highlight the item or group you want to reorder.
2. Press ALT-F and type **R** or Reorder or click on File and Reorder with the mouse.
3. Move to the new location and press ENTER or double click on the new location.

The items in the list move down to make a space for the item that you are moving.

Associating File Types with a Program

Many of the application programs you work with assign unique filename extensions to the data files you create with the program. DOS 5 allows you to associate these unique filename extensions with a specific program and will start the program for you automatically when you select any file with the filename extension you specify. You can associate as many as 20 different filename extensions with one program but each extension must be unique to the program.

As an example, you might work with 1-2-3 Release 3, on a regular basis. Since this program can read worksheet files created with Release 3 as well as earlier releases, it supports .WKS, .WK1, and .WK3 files. You might associate all three filename extensions with the program file 123.EXE. After making this association, anytime you select a file with any of these extensions, the 1-2-3 program will be started and the selected file will be retrieved to use with 1-2-3.

Establishing the Association You must use the Shell in order to establish an association between an extension and a program. You must have at least one file with the filename extension you want to use in the association in order to establish it. This file must be in the current file list. The steps you would follow are:

1. Select a filename with the desired extension.
2. Press ALT-F and type **A** or select the File and Associate menu options with the mouse.
3. Type the path and filename of the program you want to run the associated extension with.
4. Select OK or press ENTER.

If the PATH is the current directory or already included in your PATH statement you do not need to include the path in your entry.

Removing an Association If you no longer want to use an association you can remove it. The process is similar to adding it except you need to delete the program name from the Associate dialog box. Follow these steps:

1. Activate the file list that contains a file with an extension that matches the association you want to remove.
2. Highlight the filename with the association you want to remove.
3. Press ALT-F and type **A** or select File and Associate with the mouse.
4. Press BACKSPACE to delete the name and path of the associated program file.
5. Select OK.

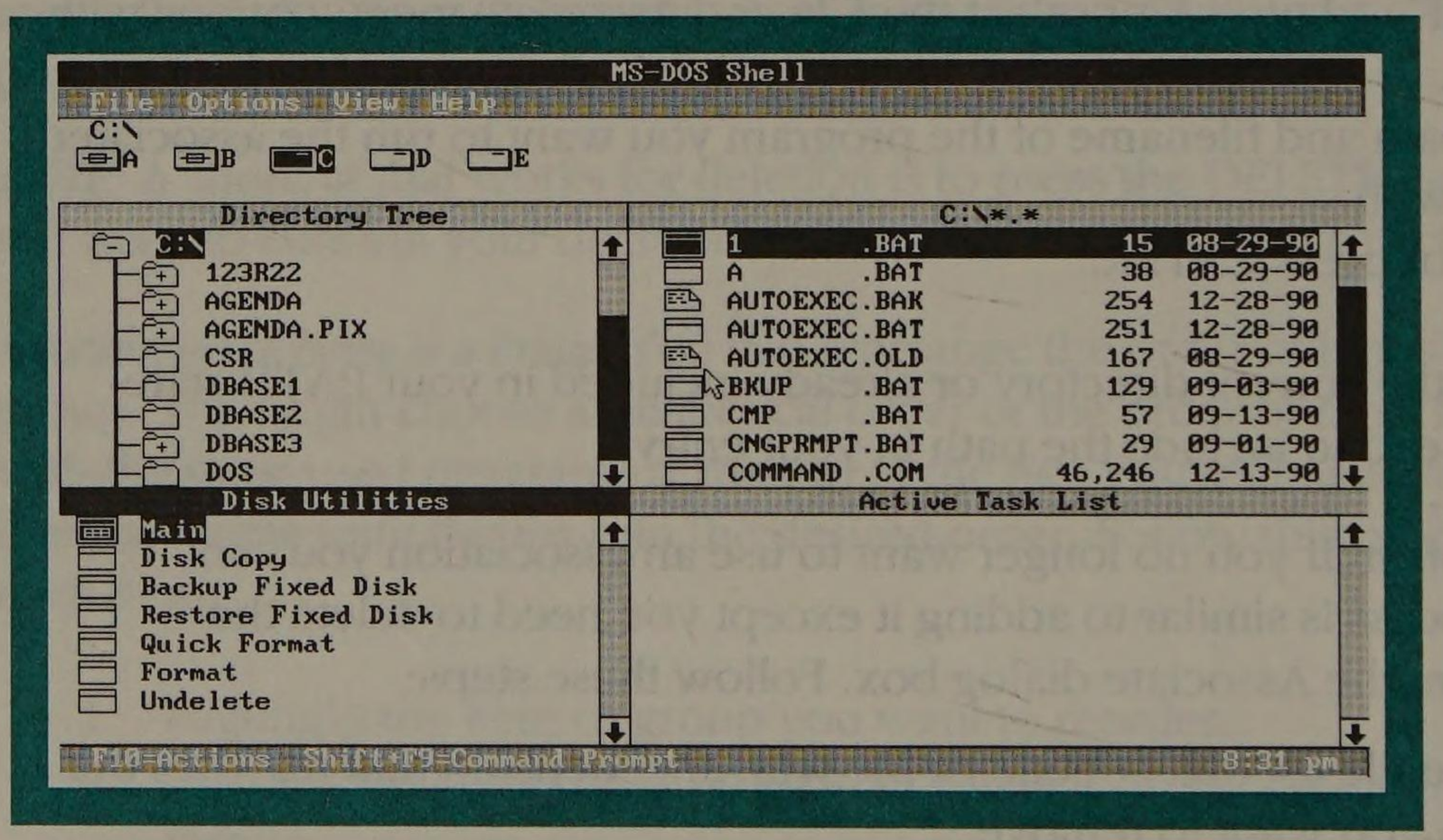
Temporarily Ignoring an Association If you want to use a file with a program other than the one that contains the association, you will need to start the program and specify the filename. If the program is part of the program list you can use the Open command. If the program is not in the program list, use the RUN command and specify the filename.

Using the Utilities

DOS provides a few disk utilities in the Disk Utilities group within the Main program list window. You can use these utilities to copy a disk, format it, undelete files, backup an entire disk, restore a complete disk, or just a section of it.

The first step in using any one of the utilities is to activate the program group Disk Utilities. With a mouse you can double-click on the Disk Utilities group name to open the group. With the keyboard, press TAB until the Main program group is active, then use the ARROW keys to highlight Disk Utilities and press ENTER. DOS displays the Disk Utilities in the bottom part of the screen as shown in Figure 5-9. You can double-click on the name of any of the utilities to start the utility program or double-click on Main to return to the Main program group.

Figure 5-9: Disk Utilities



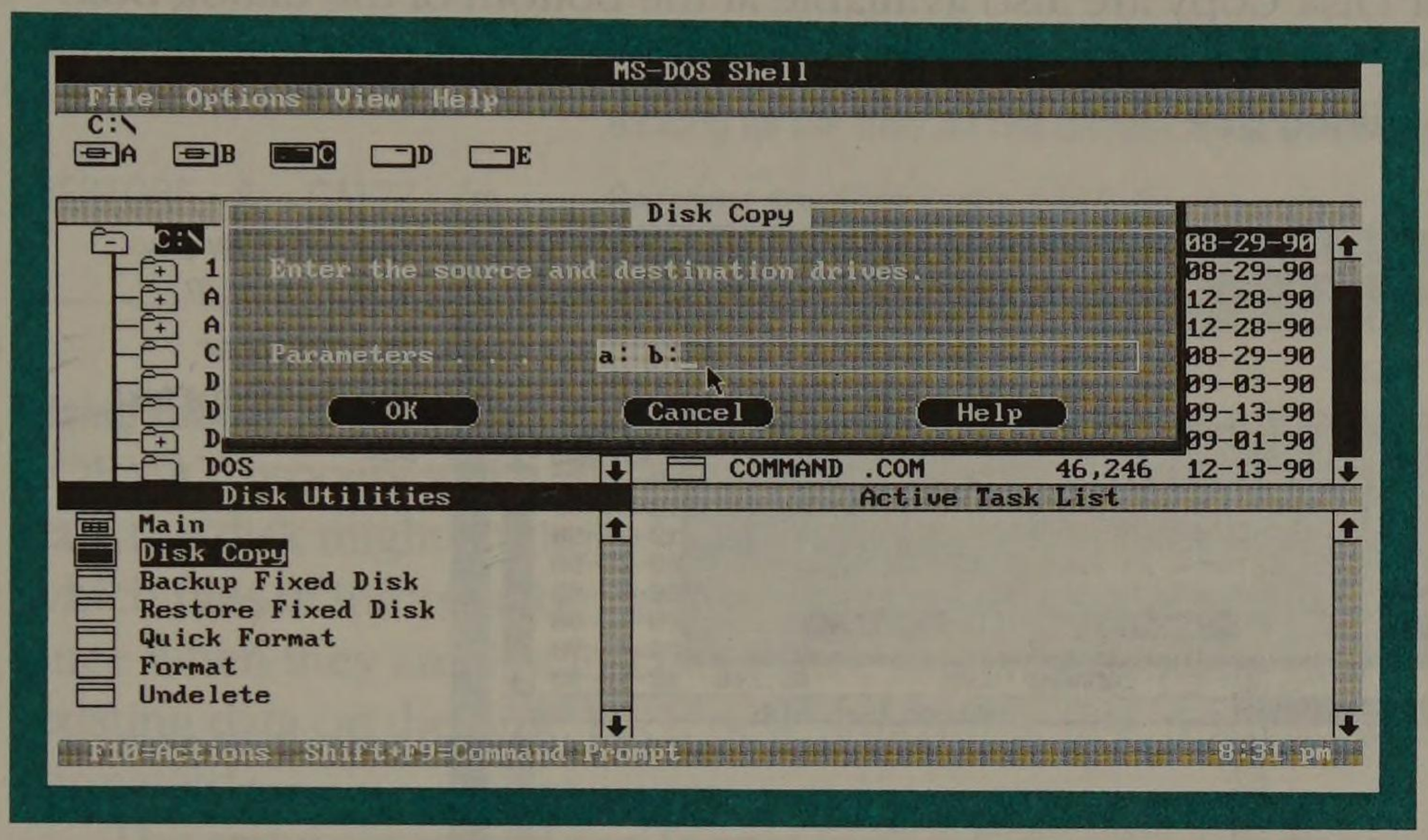
Once the Disk Utilities group is activated you can select any of the utilities.

Using Disk Copy

The Disk Copy program is designed to make a copy of a removable disk onto another removable disk of the same size and type. It offers the same features with

the DOS Shell that it does at the DOS prompt except the DOS Shell command is set up to copy disk A to disk B as shown in the default dialog box in Figure 5-10.

Figure 5-10: Disk Copy Dialog Box



Used for making a complete Disk Copy.

After selecting the Disk Copy from the Disk Utilities menu, you can press ENTER twice to make a copy of disk A on disk B. If you have a mouse, clicking on OK will perform the same task. If you want to use drive A for both the source and target disk you can press BACKSPACE twice to delete b: and type **a:** before proceeding. To copy from drive B to drive A, you would press BACKSPACE 5 times and type **b: a:** before proceeding.

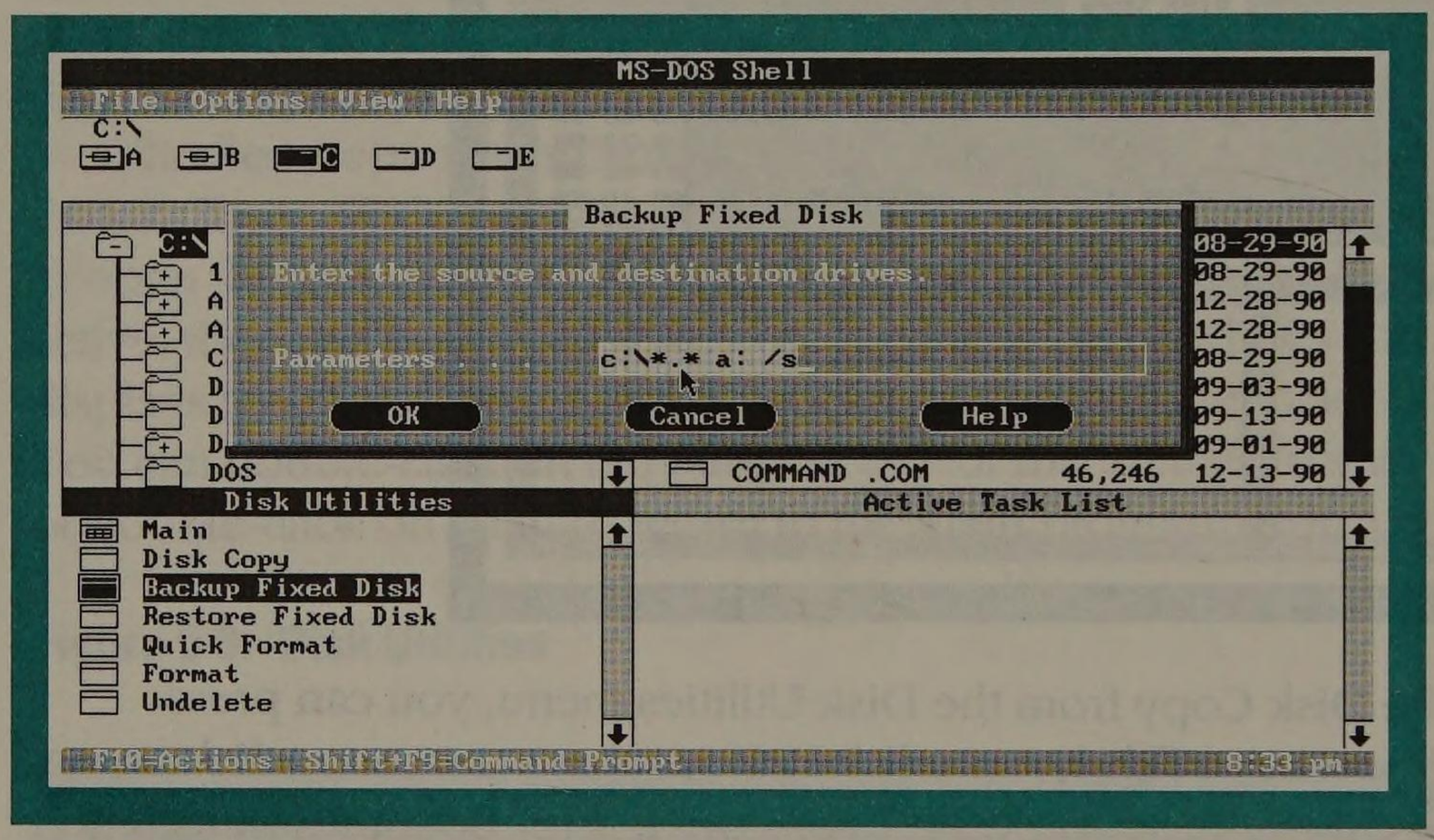
If you decide you do not want to proceed with the copy you can click on Cancel or press ESC to remove the dialog box from the screen. You can also select Help by pressing F1, pressing TAB until Help is the active option then pressing ENTER, or by double clicking on Help with the mouse.

Using Backup The Backup utility makes a special copy of files that requires less space than the COPY command, can span disks, and is used to restore a disk when its contents are lost or damaged. A regularly scheduled backup is an important part of protecting your investment in the information base you have established with your computer.

If you select Backup Fixed Disk from the Disk Utilities menu, the dialog box shown in Figure 5-11 appears. The parameter entry already supplied tells DOS to backup all files beginning in the root directory of drive C and proceeding down through all subdirectory levels and to store these backup files on drive A.

In Chapter 7, you will learn about the many different options you can use for backup. You can replace the default parameter with any valid parameter specification for the backup command. Table 5-4 shows just a few of the possibilities and the effect of the entry on the files that are backed up. The same option buttons that are available for Disk Copy are also available at the bottom of the dialog box.

Figure 5-11: Backup Dialog Box



You can enter any of the BACKUP parameters or switches in the text box.

Using Restore If you lose the data on a disk, you can restore its contents with the Restore utility as long as you have created a backup file previously for the data that you want to restore. The parameters are not automatically supplied when you make this selection since you might want to restore the entire hard disk with an entry like **RESTORE A: C:\ /s** or the 123 subdirectory with an entry like **RESTORE A: C:\123**. You can look at Chapter 7 for a complete description of all the parameter options for RESTORE although Table 5-4 provides a sampler of some of the possibilities.

Table 5-4: Sample Entries for Restore

Entry	Effect
RESTORE A: C:\WS	Restores the WS directory on drive C from the backup disk in drive A.
RESTORE A: C:\WS /s	Restores the WS directory and any subdirectories beneath it on drive C from the backup disk in drive A.

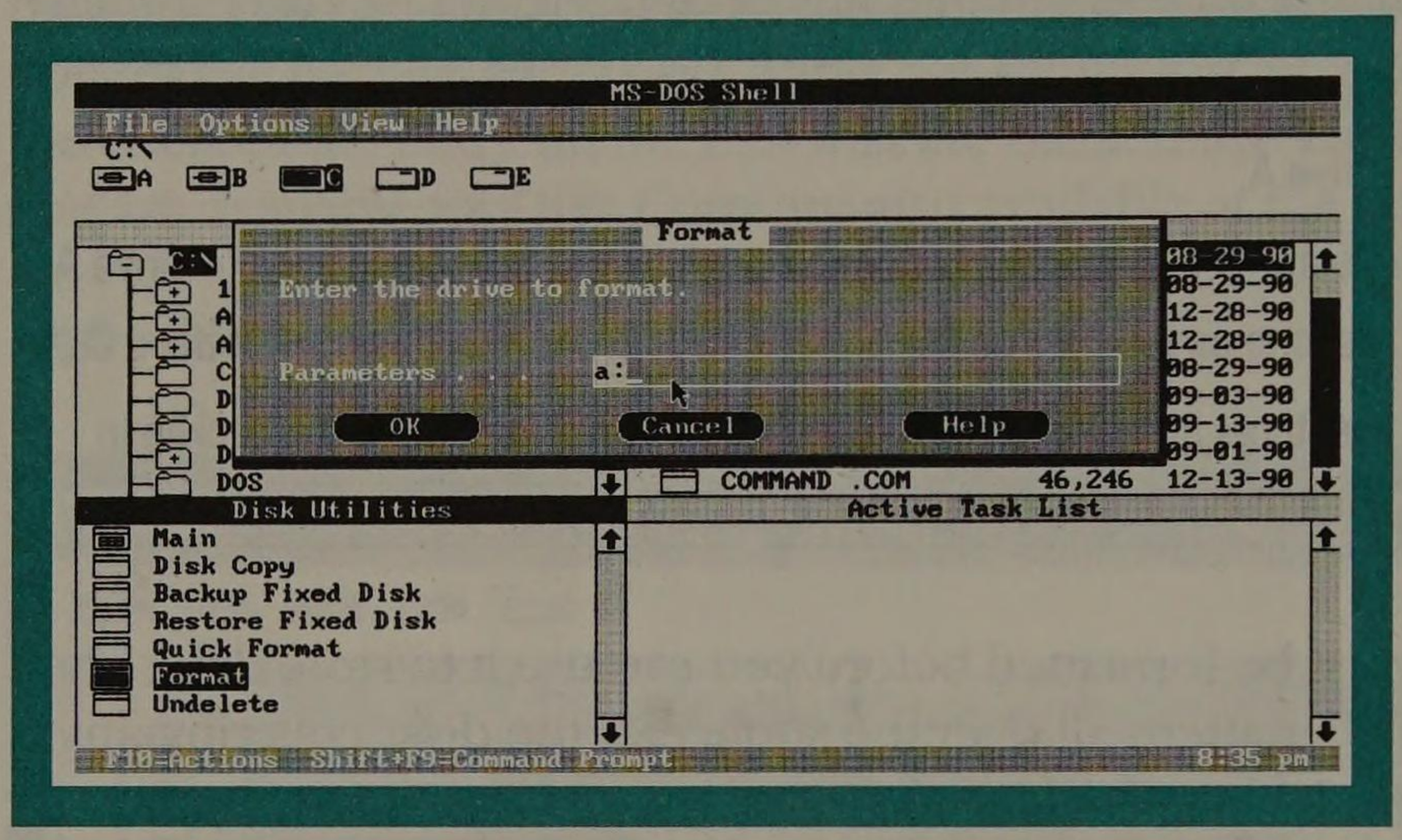
Entry	Effect
RESTORE A: C:\2.BAT	Restores 2.BAT on the root directory of drive C from the backup disk in drive A.
RESTORE A: C:\ /n	Restores files on the root directory of drive C from the backup in drive A as long as the files do not already exist in the root directory of drive C.
RESTORE A: C:\123 /m	Restores any files in the 1-2-3 directory on drive C from the backup in drive A as long as the files were modified since the last backup.

Using Format A new disk must be formatted before you can use it to store data. The FORMAT process will write a pattern all over the surface of the disk, covering any data the disk might contain. It also provides an organized writing surface upon which files of information can be stored. Hard disks are normally formatted only once when they are first installed since formatting them again would overlay the existing data on the disk. Each new removable disk that you use must be formatted.

The same command can format both 5 1/4 and 3 1/2 disks, double density and high density disks. The command will be able to determine both the size and the density from the drive you are using. To override the default density for the drive you are using, specify additional parameters. These are covered in Chapter 6 and are entered exactly as shown in the parameter text box. You have two different options for formatting a disk from the Shell. You can choose either Quick Format or Format. The Quick Format option is only useable on a previously formatted disk. Quick Format does not scan the disk for bad areas. It erases all the data on the disk by clearing the root directory and file allocation table. The Format and Quick Format dialog boxes are identical.

Initially the dialog box in Figure 5-12, which displays when you select Format, is set to format Drive A:. To accept this parameter you can either press ENTER twice or click on OK. You can format the disk in drive B by pressing BACKSPACE twice.

Figure 5-12: Dialog Box for Format



The text box for FORMAT accepts format parameter and switches.

At a Glance

In this chapter you learned about using programs with DOS and learned how to customize the Shell to fit with the programs that you need to use. You also learned how to use the disk utilities that are available through the Shell. Some of the specific tasks you learned how to perform are:

- **Enable the Switcher—File Enable Switcher**
- **Start a program—Highlight and ENTER or double-click**
- **Open a program group—File Open**
- **Close a program group—Select Main**
- **Create a new program or program group—File New**
- **Customize program settings—File Properties**
- **Switch from one application to the next—ALT-TAB**
- **Switch back to the Shell—CTRL-ESC**
- **Copy a disk—Disk Utilities Diskcopy**
- **Format a Disk—Disk Utilities Format**
- **Backup—Disk Utilities Backup**
- **Restore—Disk Utilities Restore**

Chapter 6

Managing Disk Information

Both your software and data are stored in files on your disk. Since this information is an essential resource, you will want to learn to use all the DOS commands that can help you manage your disks and the information stored on them. This chapter provides an entire toolkit of commands to put you in charge of the contents of your disks.

What You'll Learn

You will learn how to prepare disks to accept information, as well as how to make copies of them. You will learn how to eliminate data you no longer need and how to recover information that has been lost.

After reading this chapter you will be able to:

- **Prepare both floppy and removable disks for storage**
- **Create partitions and logical drives on a hard disk**
- **Duplicate the information a disk contains**
- **Verify that disks or individual files are identical**
- **Obtain status information on memory or a disk**
- **Create or display a volume label for a disk**
- **Access deleted files on a disk**
- **Work with the system information on a disk**
- **Change the names of disk files**
- **Add updated files to a disk**

Preparing Floppy Disks For Data Storage

As described in Chapter 3, floppy disks provide a permanent storage medium for your data. You cannot use disks for storage when you first remove them from their box, since they lack the formatting that tells DOS where it can place data on the disk. *Formatting* prepares the disk to store information by dividing the disk into sectors and creating a file allocation table for keeping track of the files on it.

Formatting also checks that all of the sectors can be used. If some sectors are unusable, DOS flags them in the file allocation table to prevent the use of these bad sectors later when DOS is looking for a place to store your information. If you try using an unformatted disk, DOS displays the following message:

```
General failure reading drive
Abort, Retry, Fail?
```

Using the Basic Formatting Procedure

To format a disk, use the FORMAT command followed by a reference to the drive containing the disk you want to format. To format the disk in drive A, you would enter **FORMAT A:** and to format the disk in drive B you would enter **FORMAT B:**. DOS displays a message like the one shown in Figure 6-1.

Figure 6-1: Message Indicating that Disk to be Formatted Should be Placed in Drive B

```
C:\>format b:
Insert new diskette for drive B:
and press ENTER when ready...
```

Since formatting a floppy disk eliminates any files that might exist on the disk, you should check the DOS message to make sure that the drive designation displayed is the drive that contains the disk to be formatted. If the DOS message indicates that you accidentally chose the wrong drive, press CTRL-C. If the message indicates that DOS will format the disk in the drive you want, press ENTER.

If the disk has data already stored on it and is full, DOS displays another prompt indicating that there is not enough room to create a *restore file*. The restore file contains information that allows you to restore data files if you realize that you formatted the wrong disk. To continue the formatting process, type **Y**.

Monitoring the Format Process As the disk is formatted, DOS displays that it's saving information that the UNFORMAT command will use, the capacity it is using for formatting the disk, and the percentage of the disk that has been formatted. Once the disk is formatted, DOS prompts for a label. You can enter up to 11 characters to be permanently stored as a label, or you can press ENTER to skip adding a label. A disk label helps identify the disk, just as adding an adhesive label to the outside provides an external label for the disk. Whether you choose to provide a label or omit one, DOS displays information like the text shown in Figure 6-2.

Figure 6-2: Information Displayed as FORMAT is Completed

```
C:\>format a:
Insert new diskette for drive A:
and press ENTER when ready...

Checking existing disk format
Saving UNFORMAT information
Verifying 1.44M
Format complete

Volume label (11 characters, ENTER for none)? first disk

   1457664 bytes total disk space
   1457664 bytes available on disk

       512 bytes in each allocation unit
      2847 allocation units available on disk

Volume Serial Number is 1880-16CD

Format another (Y/N)?
```

You can see the capacity of the disk, any space used by the system, the size of each allocation unit, and whether any bad areas exist on the disk.

Interpreting the Final Display from FORMAT The information DOS displays tells you about the disk it has just formatted. The total disk space indicates the number of bytes that are potentially available on the disk. Next, DOS displays the number of bytes it uses for storing system information and the number of bytes, if any, that are unavailable because they are in bad sectors. If this number is large (that is, five or more percent of the disk), you may want to throw out the disk and use another. Old disks often increase in bad sectors as they age. A large percentage of bad sectors indicates that you might be risking the data you store on the disk.

Next DOS displays the number of bytes available on the disk for your files. After this, DOS displays the number of bytes in each *allocation unit*. When DOS uses part of the disk, it uses disk space in allocation units. For example, if you

want to store a 90-byte file on the floppy disk formatted in Figure 6-2, DOS allocates 512 bytes for that file, even though the file uses only a small fraction of the file. The remainder of the allocation unit is left blank. Beneath the allocation unit size, DOS displays the number of allocation units on the disk. This number indicates how many allocation units DOS can use to store data before the disk is full. Finally, DOS displays the *volume serial number*, which is a unique disk serial number assigned to the disk until it is reformatted.

The last line displayed is a prompt to determine if you want to format another disk. You can type **N** to stop the FORMAT command or type **Y** to repeat the format process after removing the current disk and placing another disk in the same drive.

Formatting a Disk for a Lower Capacity Drive

The FORMAT command can also format disks whose storage capacity is less than the capacity of the drive you are using. This feature is useful if you are formatting a disk that will also be used on another system with a lower capacity disk drive. This feature lets you use a 1.2 MB disk drive to format 5 1/4" disks for a 360K format or a 1.44 MB disk drive to format 3 1/2" disks for a 720K format. You cannot format a disk to a capacity that the drive cannot support. For example, you cannot format a 1.44 MB disk in a 720K disk drive. This is like using a chisel when a jackhammer is needed—you waste a lot of time doing something the equipment is incapable of performing.

To format a disk in drive A to be a 360K, disk use the command:

```
FORMAT A: /F:360
```

You can also format a disk for 720K by substituting 720 for 360 in the command line and DOS displays a different format capacity as it begins formatting the disk.

■ ***NOTE:*** Due to some hardware differences between disk drives, some 360K disk drives cannot read disks formatted in a 1.2 MB disk drive. If you plan to format 360K disks in a 1.2 MB disk drive, make a trial run after formatting a disk by copying files to it and then using the disk in the 360K disk drive to make sure that the drive can read the files. Don't rely on formatting on the higher disk drive until you have performed this test. Once a disk is formatted, you can read and write data to the disk even on higher capacity disk drives.

Adding System Information to the Disk

You can also use the FORMAT command to transfer system files that DOS uses to boot the computer. This is useful when you are creating a disk that you want to use to start the system.

To add the essential system information to a disk, use the command:

```
FORMAT A: /S
```

The rest of the FORMAT command is the same. You may need to put a system disk in drive A if you did not boot the computer from the hard disk. The /S switch tells DOS to add the three system files after the disk is formatted. These files are IO.SYS, MSDOS.SYS and COMMAND.COM. The first two are hidden, so you will not see them when you look at a drive's contents. After formatting a disk to be a system disk, you will notice that the number of bytes taken by the system has increased. Later, when you boot your computer, it first checks drive A to determine if this drive contains a system disk before it checks the hard disk.

■ ***NOTE:*** Formatted disks with the system files on them are particularly useful for users without a hard disk. These users can copy a program or data to the system disk and start their system without having to change the disk in the disk drive.

If you later decide that you want a disk to be a system disk, you can make it a system disk by entering these two commands:

```
SYS A:
COPY \COMMAND.COM A:
```

The first command transfers the two hidden system files to drive A and the second command copies the COMMAND.COM file from the root directory of the current drive to drive A. This last command is covered in more detail in Chapter 7. Besides the /S switch, the FORMAT command also has a /B switch, which copies the two hidden system files without copying the COMMAND.COM file. This switch is useful when you want to create a boot disk that uses a different command processor than COMMAND.COM (a very advanced feature).

Other FORMAT Command Switches

The FORMAT command has other switches that you can use to control how DOS formats your disk. You can use the /U switch to format the disk so it cannot later be unformatted. You can use the /Q switch to quickly format a disk. This switch

does not erase the information already on the disk or check for bad sectors, but erases the file allocation table and directory information. This switch produces the same results as Quick Format in the Disk Utilities program group of the DOS Shell.

You can use /V: followed by the text to assign the volume label, thus pre-empting the prompt for the volume label during the format command. The /1 switch formats only one side of the disk. The /1 switch can also be combined with the /4 switch or /8 switch to format a 5 1/4" disk with a low format. By itself, the /4 switch formats a 5 1/4" disk for 360K; when used with the /1 switch, /4 formats the disk to 180K. The /8 switch formats the disk for 8 sectors instead of the default of 9 used by DOS 2.0 or above. The /F switch is the replacement for the /N and /T switches. When you use /F: followed by the disk capacity such as 360, 720, or 1.2 MB, DOS automatically determines the number of sectors and tracks to use for formatting. When you use the /N and /T switches you must follow them with colons and the number of sectors per track and the number of tracks on the disk.

Preparing a New Hard Disk

When you add a hard disk to your system or want to change how an existing one is set up, you need to make several changes. First, you need to decide how to structure your hard disk storage. You can treat the entire disk as one repository for data, or you can partition it into different sections in the same way that a large theater can be partitioned into a number of smaller rooms. *Partitions* are divisions of a hard disk that can be used as different logical drives or can be used for different operating systems. It is essential to perform the partitioning step *before* formatting a hard disk. After a disk is partitioned, you can format the logical drives that the disk uses.

Establishing Partitions

In Appendix A you can learn how to partition a hard disk during the installation of DOS. You can use different partitions to run different operating systems such as DOS, OS/2 and XENIX. You can create as many as four different partitions, although only one may be a primary DOS partition.

You can also partition a hard disk with the FDISK command. If you change the partitions, you will want to do this before you copy programs and data files to a hard disk, since after you create a partition, you must format it. The only uses of the FDISK command without changing partitions are to display the current partition information or to change which partition is active (the partition that the computer uses to boot from). While you are in the FDISK command menu, you can select menu items by typing the number next to the menu item you want and pressing ENTER, or you can return to the prior menu (or return to the DOS com-

mand prompt) by pressing ESC. If you have altered the DOS partition sizes while using FDISK, DOS displays the message **System will now restart**. If you have altered the active primary DOS partition, you must put a DOS disk in drive A.

You can quickly display the partition information by typing **4** from the FDISK menu to select Display partition information, and pressing ENTER. This displays information like the text shown in Figure 6-3. This information will vary according to your disk capacity and how it is divided. The columns show the hard disk letter, the number of partitions it is divided into, which one is active, the type of partition (primary DOS, extended DOS or non-DOS), the number of megabytes, and the percentage of disk space each partition uses.

Partitions can be further divided into *logical drives* to allow you to refer to different parts of the physical drive by different drive letters, such as drive D, drive E or drive F. If a partition is divided into logical drives, you can display that information by typing **Y** in response to the prompt and pressing ENTER to display the logical drive information like the one shown in Figure 6-4. This information includes the volume labels assigned to the logical drives with the format command, the number of megabytes in each, the system each logical drive is designed to use and the percentage of the disk each logical drive uses.

Figure 6-3: Partition Information

```
                    Display Partition Information

Current fixed disk drive: 1

Partition Status   Type    Size in Mbytes    Percentage of Disk Used
 C: 1        A    PRI DOS        32              75%
    2             EXT DOS        10              25%



Total disk space is    42 Mbytes (1 Mbyte = 1048576 bytes)


The Extended DOS Partition contains Logical DOS Drives.
Do you want to display the logical drive information (Y/N).....?[Y]



Press Esc to return to FDISK Options
```

This example shows a 42 MB hard drive divided into a 32 MB primary DOS partition and a 10MB extended DOS partition.

If you have more than one hard disk, you may need to change the disk that DOS assumes you want to use with the FDISK command. DOS initially assumes that the first hard disk in the computer is the one you want to use. To work with another one, type **5** from the FDISK menu to select Change Current Fixed Disk Drive and press ENTER. Then type the number of the hard disk you want to change to and press ENTER. If you have only one fixed disk, the Change Current Fixed Disk Drive option does not appear on the FDISK menu.

Figure 6-4: Logical Drive Information

```
                Display Logical DOS Drive Information

Drv Volume Label  Mbytes  System  Usage
D:  METRO             1  FAT12     10%
E:  DATA FILES        5  FAT12     45%
F:  PROGRAMFIL        5  FAT12     45%
```

The 10 MB extended DOS partition in Figure 6-3 is divided into three logical drives.

Deleting a Partition or Logical Drive DOS does not let you edit the partitions, so if you want to change a partition's size, you must delete it and then recreate it. Deleting a partition permanently eliminates the ability to access the information that was once stored in that partition. Deleting a partition has no effect on the data stored in another partition. Before you can delete a primary DOS partition on a hard disk, you must delete any logical drives and extended DOS partitions on the disk. You must also delete any logical drives on an extended DOS partition before deleting the extended partition.

To remove a DOS partition or a logical drive, type **3** from the FDISK menu to select Delete DOS Partition or Logical Drive and press ENTER. Then type **1** to select Delete Primary DOS Partition, **2** to select Delete Extended DOS Partition, **3** to select Delete Logical Drive(s) in the Extended DOS Partition, or **4** to select Delete Non-DOS Partition.

After you select one of these, DOS displays a list of the partitions or logical drives to select from along with a warning that data in the partition will be lost. Type the partition number or logical drive letter that you want to delete and press ENTER. For logical drives, you must also enter the volume label and press ENTER. This double checks that DOS deletes the logical drive you want. Finally, DOS prompts for an additional confirmation. You must type **Y** and press ENTER to confirm that you are sure you are deleting the logical drive or partition you have selected. Then press ESC to return to the FDISK menu. When you leave the FDISK menu, DOS restarts the system.

Creating a Primary DOS Partition The primary DOS partition is the partition that DOS boots from. If you have multiple hard drives, it must be on the first hard drive. You can use your entire disk or only a portion for this partition. If you have an existing primary DOS partition, you will want to delete it before creating a new one. To create a primary partition, type **1** from the FDISK menu to select Create DOS Partition or Logical Drive and press ENTER. Then type **1** again to select Create Primary DOS Partition. DOS prompts if you want to use the entire drive for the primary partition and make it the active one. You can type **Y** to use your entire hard drive for a partition, or you can type **N** if you want to divide a hard drive into smaller sections. If you type **Y**, you can put the Install disk in drive A and press any key. If you type **N**, DOS prompts you for the number of megabytes or the percentage that you want to use for the primary partition. Either type the number of megabytes or type the percentage followed by a percent sign (%) and press ENTER. When DOS displays a message that the primary DOS partition is created, press ESC to return to the FDISK main menu. At this point, you will probably want to create an extended DOS partition with the remaining capacity of your disk.

Unless your sole use of FDISK has been to create a primary DOS partition for the entire disk, you must now select the *active* partition. To make a partition the active partition, type **2** to select Set Active Partition and press ENTER. DOS displays a list of the partitions, with an A indicating the currently selected or active partition. Type the number of the partition you want to change. (DOS will not accept the change if you select an extended DOS partition.) Then press ESC to return to the Main Menu. When you reboot the system, DOS will boot from the active partition. This is the procedure you would use to make a non-DOS partition, such as one created for UNIX, the active partition so you can boot to that operating system.

Creating an Extended DOS Partition *Extended* DOS partitions are disk areas DOS can use that are not part of the primary partition. To differentiate the partitions, DOS assigns unique drive letters to each partition. You can also divide an extended partition into several logical disk drives, which DOS treats as if they were separate physical drives.

To create an extended partition, type **1** from the FDISK menu to select Create DOS Partition or Logical DOS Drive and press ENTER. Then type **2** to select Create Extended DOS Partition. DOS displays the remaining number of megabytes that are available for an extended DOS partition. The default is to use all the available space for the extended partition you are creating. You can press ENTER to use all the available space for the extended partition, or you can type the number of

megabytes or the percentage that you want to use (followed by a %) and press ENTER. After the partition is created, press ESC.

Next, DOS displays the Create Logical Drive(s) in the Extended DOS Partition menu. DOS prompts you for the number of megabytes or the percentage that you want to use for the logical drive. Either type the number of megabytes or type the percentage followed by a %. You can also use the default of 100% by just pressing ENTER. When the number you want is displayed, press ENTER. If you have additional megabytes left unassigned to logical drives, you can repeat the process to select a percentage or number of megabytes for another logical drive. When all of the extended partition is assigned to logical drives, DOS automatically returns to the main FDISK menu. Once these logical drives are created, you can format them later.

The primary partition and logical drives you create are assigned unique letters. Assuming you have only one or two floppy disks, the primary partition is drive C. The extended partitions are assigned subsequent letters (up to Z) according to the order in which they are created in the FDISK program. If you have three or four floppy disk drives, the primary partition is drive D or E, with the logical drives using the letters after that. If you have primary partitions on multiple disks, DOS assigns letters first to all primary DOS partitions in the order in which the hard disks are attached and then to any logical drives.

Formatting the Hard Disk

Once a hard disk is partitioned, you are ready to format it to prepare it to store your data. You may also want to reformat a hard drive when you have used it for a long time and want to erase the disk and then copy the files back to the directory. (Warning: make sure you back up all of your needed files, as described in Chapter 7, before beginning the reformat operation.) Another reason for reformatting is because files are not necessarily put into contiguous sectors as the disk is used. Thus, as files grow larger and use more and more free sectors that are spread out over all areas of the disk, they become fragmented. When this happens, DOS must take longer to read a file since it must jump around to find each sector where a piece of the file is stored. This increases the time DOS and other applications take to work with your files.

To format a hard disk, you use the same FORMAT command you used to format floppy disks. To format a hard disk, type the FORMAT command with the drive letter, as in:

```
FORMAT C: /S
```

You will need to use the /S switch when you format the primary DOS partition so that DOS will transfer the system files to the partition.

When you format a hard drive, DOS displays an extra warning to prevent you from unintentionally formatting the wrong disk. Also, if the disk is already formatted, you must enter the correct volume name before DOS can proceed. The FORMAT command must be repeated for every DOS partition. You can check which disks need to be formatted by using the CHKDSK command described later in this chapter. If the disk has not been formatted, DOS displays a message, **Probable non-DOS disk, Continue**. You can type **N** to halt the command.

When you format the drives, you may want to provide volume labels using the same procedures described in the section on formatting floppy disks earlier in this chapter. These volume labels are the same labels you will see when you use the FDISK command to display the current partition and logical drive information.

Copying a Disk

DOS contains several commands that let you quickly make duplicates of files on your disk. One of the simplest, the DISKCOPY command, makes a replica of a disk on another disk of the same capacity. You enter the command like this:

```
DISKCOPY A: B:
```

DOS prompts you to place the *source* (the disk that you want to copy) and *target* (the disk on which you want to store the new copy) disks into the selected drives and press any key. DOS then copies the contents of the disk in the first selected drive to the disk in the second selected drive. Any previous contents of the target disk are erased. If the target disk is not yet formatted, DOS formats the disk as it copies, and displays the disk serial number of the new copy of the disk (it will be different than the original). Once the disk is copied, DOS will prompt if you want to copy another diskette (using the same source and target drives). Type **Y** to copy another disk or **N** to halt the DISKCOPY command.

You can also use the same drive for both the source and target, as in:

```
DISKCOPY A: A:
```

In this case, DOS prompts you to insert the source disk in drive A, and after it finishes reading data from the source disk, prompts you to enter the target disk in the same drive. If you omit the source and target drives, DOS assumes that the source and target drives are the current ones. DOS will not let you use a hard drive as the source or target drive for this command.

The DISKCOPY command has two options. You can use /1 to copy only the first side of the source disk. This is used with single-sided floppies. You can also use the /V switch, which directs DOS to double check itself that it has copied the disk correctly. If you use this switch, the command takes longer. Alternatively, you can make the DISKCOPY command, and all DOS commands that write to the disk, verify that files are written correctly by entering the VERIFY command like this:

```
VERIFY ON
```

This causes DOS to double check itself when it writes data to the disk with commands like DISKCOPY and COPY (introduced in Chapter 7). This command remains in effect until you enter VERIFY OFF at the DOS command prompt.

Comparing Disks

DOS has three commands for comparing files. These commands let you confirm that files are identical. You can use these commands to make sure that a backup disk matches the original and to make sure that the same files on floppies are the same. This is especially important if your computer does not permanently keep track of the date and time and you forget to provide the date and time when you boot the computer. The three commands are DISKCOMP, COMP and FC. The command you use depends on whether you want to compare all of the files on two disks or if you only wish to compare one or more files.

Comparing Two Disks

When you have two disks that you think contain the same data, you may want to verify that the disks are identical. DOS can perform this comparison for you with the DISKCOMP command. You can enter the command like this:

```
DISKCOMP A: B:
```

This assumes the disks you want to compare are in drives A and B. Like the DISKCOPY command, you can use the same drive for both disks (or provide no drive to use the current one). If the two disks are identical, DOS displays the message **Compare OK**.

If the disks are not the same, DOS displays an error message, showing the first track number and side number that differed.

Also, like the DISKCOPY command, when DISKCOMP finishes comparing two disks, it prompts whether you want to compare another two disks. You can type **Y** to repeat the DISKCOMP command for another set of disks or type **N** to end the

DISKCOMP command. If the drive you booted with, is the drive you used for the DISKCOMP command, or if you compare 1.2 MB or 720K disks, the DISKCOMP command may prompt you to put the disk with the COMMAND.COM file in drive A and strike a key. The DISKCOMP command requires that the source and target disks be the same format, which means you cannot compare a 1.2 MB 5 1/4 inch disk to a 360K 5 1/4 inch disk. This command has two switches, /1 and /8, which are used, respectively, when you are comparing single-sided disks or disks that have eight sectors to the track (160K and 320K disks only). This command should not be used on copies of a disk made by copying individual files, as the COPY and XCOPY commands (discussed in Chapter 7) do. If you back up a disk using these commands, you should instead use the FC or COMP commands to compare its files to those of the original disk.

Comparing Files with the COMP Command

The COMP command compares two files and reports on any differences. Usually, you use this command by entering:

```
COMP FILE1 FILE2
```

This compares the data in FILE1 with the data in FILE2. The filenames can be the same. This lets you compare files on different directories and drives. When you enter this command, DOS compares the two files byte by byte. If the two files are the same, DOS displays the message **Files compare OK**. It then prompts you whether you want to compare additional files.

You have many options for entering the pathnames of files to compare. For example, you can use the question mark and asterisk wild cards to select files. DOS assumes that if you provide a drive and/or directory with no filename after it, the filename is *.*, and therefore it compares all of the files in the directory. If you do not provide the filenames to compare, DOS prompts for them, along with any options you may want to use with the COMP command, as described below. DOS only compares the files selected by the first filename parameter and ignores any extra files selected by the second filename parameter.

When DOS finds a difference between the first and the second file, it creates a display like the one in Figure 6-5. After ten differences, DOS stops comparing the two files and displays a message. If the files are different sizes, DOS displays a message that the files match even though they are different sizes. You can either type **N** to end the COMP command or **Y** to compare other files.

The COMP command has several switches you may want to use. You can use the /d switch to display differences between two files using a decimal format (the

default is to use hexadecimal—base 16). You can also use the /a switch to display the differences as characters. You cannot combine /d with /a. You can display the line number containing the difference instead of the number of bytes from the beginning of the file by including /l. You can also select the number of lines to be compared starting from the beginning of the file by entering /n: followed by the number of lines you want to compare. Finally, you can use /c to make COMP ignore differences caused by the same letters in upper and lowercase.

Figure 6-5: Display Indicating Difference in Two Files During Comparison

```
C:\WP51>comp notes2 b:
Comparing NOTES2 and B:NOTES2...
Compare error at OFFSET 4
file1 = 70
file2 = 66
Compare error at OFFSET 5
file1 = 72
file2 = 69
Compare error at OFFSET 6
file1 = 6F
file2 = 6E
Compare error at OFFSET 7
file1 = 64
file2 = 61
Compare error at OFFSET 8
file1 = 75
file2 = 6E
Compare error at OFFSET A
file1 = 74
file2 = 65
Compare more files (Y/N) ? n

C:\WP51>
```

The output of the COMP command displays each different character or byte separately. This makes comparing text files hard to read.

Comparing Files with the FC Command

Like the COMP command, the FC command compares two files and reports any differences. The FC command differs from the COMP command, however, in several respects. First, when a difference is found between the two files, the FC command tries to resynchronize the files. This is especially important in word processing files, where an added word will shift all of the subsequent characters. The FC command figures out where the two files return to being the same after the original difference, while the COMP command simply continues reporting differences until it finds ten differences, and then stops. The output of the FC

command is also easier to understand, since it displays the context surrounding each difference to allow you to find differences in files quickly.

Usually, you use the FC command by entering:

```
FC FILE1 FILE2
```

This compares the data in FILE1 with the data in FILE2. The filenames can be the same. This lets you compare files on different directories and drives.

When you enter the FC command, DOS compares the two files. If the two files are the same, DOS displays a message that no differences were found.

As with the COMP command, you have several options for entering the pathnames of the files to compare. For example you can use the question mark and asterisk wild cards to select files. Unlike the COMP command, however, you must include the filename after the path information even if it is *.* Also unlike the COMP command, you cannot enter FC without parameters. DOS will not prompt you for the information it needs. DOS ignores additional files selected with the second filename parameter that are not selected with the first.

The FC command produces different output than the COMP command. When FC finds a difference between the first and the second file, it produces output like that shown in Figure 6-6. This output displays the first filename, the line before the differences occur, the lines containing the differences, the first lines that match in both files, and the same information for the second file. Unlike COMP, FC continues finding differences until it reaches the end of one of the files.

Figure 6-6: Sample Output of the FC Command

```
C:\WP51>fc notes2 b:notes2
Comparing files NOTES2 and B:NOTES2
***** NOTES2
The product meeting is tomorrow.
Prepare proposals for distribution at the meeting.
***** B:NOTES2
The finance meeting is tomorrow.
Prepare proposals for distribution at the meeting.
*****


C:\WP51>
```

The output of the FC command is easy to read when you are comparing text files.

This command has several switches you may want to use. You can use the /b switch to perform a binary comparison of the two files. A binary comparison does not rematch the two files, and the output of the differences will contain the hexadecimal address within the file that contains the differences and the characters in that position in each file. You can also use the /l switch to make an ASCII comparison, assuming the data contains text. This is the default for all files except those with extensions of .EXE, .COM, .SYS, .OBJ, .LIB, and .BIN which use a default of a binary comparison.

If the files use tabs, you may want to use the /t switch to prevent DOS from expanding tabs to eight spaces. You can also have the comparison ignore the effect of extra spaces and tabs by using the /w switch, which treats multiple spaces, tabs and combinations as if they were a single space.

You can display the line number containing a difference by including the /n switch. You can shorten the display to include only the two filenames and the beginning of the first and last lines that contain differences using the /a switch.

You can also select the number of lines that DOS temporarily stores to resynchronize the lines that match between the two files. Normally, DOS remembers one hundred lines of each file, so if ninety-eight lines have been inserted into one of the files, DOS can use the last two lines in its buffer to figure out where the two files start being back in synchronization again. To change the number of lines, enter /lb followed by the number of lines. If DOS cannot resynchronize the two files, it displays a message that the Resynch failed. The FC command assumes that two files are back in synchronization when two contiguous lines from the two files match. You can change this from the default of two to another number by typing / followed by the number of contiguous lines in each file that must match for the two files to be in synchronization again.

Finally, you can use /c to tell FC to ignore differences caused by the same letters in upper and lowercase.

Working with Disk Labels

Earlier in this chapter, you saw how you can add volume labels and serial numbers to a disk when you format it. DOS uses the volume serial number to check that the drive still contains the disk that was last in the drive. By comparing the volume serial number last used to the current one in a drive, DOS knows when you have switched disks. The disk labels are up to eleven characters and cannot include tabs, control characters (characters created by pressing the CTRL key and another key) or any of the following symbols:

*	.	=	?	,	/	:	[	]	\	;	|	+	&	(	)	<	>	'

Adding a Label

Besides adding a label when you format a disk, you can add a label with the LABEL command at a later time. This command is entered like this:

```
LABEL A:FIRST DISK
```

This labels drive A with the label FIRST DISK. Any lowercase letters are automatically converted into uppercase letters. The entry after the drive letter and colon must follow the rules given above for label names.

You can also simply enter:

```
LABEL A:
```

In this case, DOS displays the current label and serial number (if any) for the disk in the designated drive and prompts for a new label. You can also use this format of the LABEL command to remove a label, by pressing ENTER when DOS prompts for the new label name. If you are deleting the volume label from a hard disk, DOS will prompt you to confirm that you want to delete the volume label. Type **Y** to change the label or **N** to halt the LABEL command.

Looking at a Disk Label

When a disk has a label, DOS displays the volume label and volume serial number at the beginning of the DIR command (covered in Chapter 3). You can also display the label with the VOL command. When you enter VOL and press ENTER, DOS displays the volume label and serial number of the current disk. If you follow VOL by a drive designation, as in VOL A:, DOS displays the volume label and volume serial number of the disk in the selected drive. DOS also displays the volume label and volume serial number during the CHKDSK command described below.

Checking the Status of a Disk

DOS includes a CHKDSK command that checks and reports general information about a disk. This command checks the file allocation table and file system for any logical errors, such as two or more files that use the same allocation unit or cluster. It also finds allocation units that are not marked as available for storing data but do not have directory entries that identify them. The CHKDSK command is entered like this:

```
CHKDSK A: /F /V
```

This example checks the drive on disk A. If it finds any logical errors, it corrects them since the /F switch is included. The /V switch, for example, lists the names of the files as they are checked on the disk. These two switches are optional. The output of the CHKDSK command is shown in Figure 6-7. Another option is to enter CHKDSK followed by a file name or file name specification (such as *.*). Besides the usual CHKDSK output, this command will list the fragmented files and the number of non-contiguous blocks used to store each file.

Besides using the CHKDSK command to check on the status of a disk and the available RAM, you can also use the MEM command. When you use the MEM command, DOS displays the number of bytes in memory, the number of programs that are in memory, and the list of programs.

Using CHKDSK's Output

The output the CHKDSK command produces provides you with different information about the disk. Using CHKDSK you can look at the volume name, when it was created (or formatted), any volume serial number, any files listed (if /V is used), any errors, and a status report. Figure 6-7 shows one of these status reports. As you can see in the figure, DOS displays the number of bytes that are on the disk, the number of bytes used by hidden files, the number of bytes used by non-hidden files, the number of bytes available for storing more files, the number of bytes in each allocation unit, the number of allocation units on the disk and the number of allocation units available for storing more files. Finally, the status report displays the amount of conventional RAM and the available RAM that you can use for other applications. With this information, you can see how much room remains on the disk, as well as how many hidden and non-hidden files the disk contains. If you try running a program and get a message that there is not enough memory, you can use the CHKDSK command to check the amount of RAM available for programs.

Taking a More Detailed Look at the Disk Status

If DOS finds lost allocation units, it indicates how many allocation units are unaccounted for and in how many chains these allocation units are grouped. Several clusters are part of a chain when one cluster refers to another until a cluster no longer ends with another cluster reference. This is why DOS normally only records the first cluster of a file, since the other clusters that contain part of the file's information are chained together. If you did not use the /F switch, DOS displays a message like this:

```
Errors found F parameter not specified
Corrections will not be written to disk
7 lost allocation units found in 2 chains.
7168 bytes would be freed
```

If you used the /F switch, you are prompted whether you want to convert these chains into files. If you type **Y**, DOS saves each chain in a separate cluster in the root directory. These files have names like FILE0000.CHK, FILE0001.CHK and so forth, which are in the root directory. After you create these chains, you can review the contents and erase the files if you do not want them. If you type **N**, DOS adjusts the file allocation table for the logical errors by making the allocation units that it could not account for, available for storing new files.

Figure 6-7: Output from CHKDSK

```
C:\WP51>chkdsk a: /f /v
Volume Serial Number is 0D69-1ACF
Directory A:\
A:\IO.SYS
A:\MSDOS.SYS
A:\COMMAND.COM
Directory A:\FILES
A:\FILES\FILE_1.TXT
A:\FILE_1.TXT

   1457664 bytes total disk space
     87040 bytes in 3 hidden files
       512 bytes in 1 directories
     43008 bytes in 3 user files
   1327104 bytes available on disk

       512 bytes in each allocation unit
      2847 total allocation units on disk
      2592 available allocation units on disk

    655360 total bytes memory
    583136 bytes free

C:\WP51>
```

This output shows the files on the disk since you used the /v switch. The remaining information displays different statistics about the disk and conventional RAM.

■ ***NOTE:*** If the disk contains several files that use the same cluster, you may need to use a program such as Norton's Utilities to look at the cluster or allocation unit, determine which of the files it belongs to, and then delete the other files.

Using MEM to List Memory Usage

You can use the MEM command to display available memory, the programs in memory, internal drivers, and other programming information. When you type MEM and press ENTER, DOS displays the number of bytes of conventional memory, conventional memory available to DOS, conventional memory available for executing another program, expanded memory, and extended memory as well as where DOS is loaded. This command only displays information about expanded and extended memory if you have that memory type installed on your computer. Figure 6-8 shows the output of using the MEM command with no switches.

Figure 6-8: Ouput from the MEM Command

```
C:\WP51>mem


    655360 bytes total conventional memory
    654336 bytes available to MS-DOS
    583152 largest executable program size

   2490368 bytes total contiguous extended memory
         0 bytes available contiguous extended memory
   2424832 bytes available XMS memory
           MS-DOS resident in High Memory Area

C:\WP51>
```

The output of the MEM command provides an overview of memory allocation and availability.

The MEM command has two switches that you can use to display information about how DOS uses the memory. If you use the /PROGRAM switch, DOS displays the starting memory address, name, size and type for programs and other applications DOS uses part of RAM for. You can use this to check where DOS loads programs in memory as well as how much room DOS uses for part of the operating system. Figure 6-9 shows a sample of this output. Using the /DEBUG switch provides the same information, but it also includes more information about the internal drivers (files DOS uses to run your hardware, such as your mouse). Both of these switches are primarily used for advanced programming applications, but you may want to use MEM without switches to see how much RAM you have in the different types of memory.

Figure 6-9: Using MEM /PROGRAM to Display How DOS Uses Memory

Address	Name	Size	Type
000000		000400	Interrupt Vector
000400		000100	ROM Communication Area
000500		000200	DOS Communication Area
000700	IO	000AE0	System Data
0011E0	MSDOS	001500	System Data
0026E0	IO	005230	System Data
	MOUSE	002A00	
	ANSI	001180	DEVICE=
	HIMEM	000410	DEVICE=
		0005D0	FILES=
		000100	FCBS=
		000200	BUFFERS=
		000210	LASTDRIVE=
		000740	STACKS=
007920	MEM	000060	Environment
007990	MSDOS	000010	-- Free --
0079B0	MSDOS	000040	System Program
007A00	COMMAND	000930	Program
008340	MSDOS	000040	-- Free --
008390	COMMAND	000100	Environment
0084A0	APPEND	002070	Program
00A520	GRAPHICS	0016C0	Program
00BBF0	MEM	013480	Program
01F080	MSDOS	080B70	-- Free --

655360 bytes total conventional memory

654336 bytes available to MS-DOS

606208 largest executable program size

2490368 bytes total contiguous extended memory

0 bytes available contiguous extended memory

2424832 bytes available XMS memory

MS-DOS resident in High Memory Area

Erasing Files from the Disk

As you create new files on a disk, you will sometimes want to make space by removing the ones you do not want. You can erase them one at a time, or in groups, with the DEL or ERASE command. If you accidentally erase a file you do not want deleted, DOS includes a command to restore these files.

When you delete a file with DOS, you do not actually erase the data. Instead, DOS changes the first character of the filename to a special character reserved to indicate deleted files. DOS also changes the file allocation table for the disk so that the disk space that the file used is marked as available for new files. Since DOS uses this available area to store the next file you add to the disk, it is important that you undelete a file as soon as possible if you change your mind.

To delete a file, type **DEL** and the filename, as in:

```
DEL MY_FILE
```

(Note: You can also use ERASE in place of DEL.) This deletes the MY_FILE file from the current directory. If the file is in another directory or drive, you must include that information, as in DEL A:\FILES\MY_FILE. You must also include any file extension the file has, so that DOS knows you want to delete MY_FILE.DAT instead of MY_FILE.

Another option you may want to consider is using the ? or * wildcards. For example, you can delete all of the files on drive A by entering:

```
DEL A:*.*
```

Since you can delete all of the files with this command, DOS displays a warning that all of the files in the directory will be deleted. You must type **Y** to delete these files. When you delete all of the files in a directory, DOS does not delete any subdirectories the current directory has, any hidden files or any files that are read-only (an attribute that you can add as described in Chapter 3). When you use the wildcard characters, you may also want to use the /P switch, which causes DOS to prompt for each file whether or not you want to delete the file. You can delete the file by typing **Y** or skip deleting the file by typing **N**.

Tracking the Information on a Disk

DOS keeps track of the files on your disk and where they are located through the file allocation table and the root directory. Besides using these as sources of information, you can also back-up this information so that if the directory ever becomes damaged you can restore it. You can also track the files that you delete

and where they are stored so that you can use the UNDELETE command (described later in this chapter) to recover them. The MIRROR command can duplicate the file storage information, monitor the files you delete or make a special copy of the partition information. Once the MIRROR command is used, you can use the UNFORMAT command to recreate a directory, UNDELETE to recover part or all of a deleted file or UNFORMAT to recreate a partition table so you can access a hard drive.

Using MIRROR to Copy File Storage Information To use the MIRROR command to duplicate file storage information, enter MIRROR followed by the drive or drives from which you want to record the file storage information. Examples of the MIRROR command used in this way are MIRROR C: or MIRROR A: B: C:. If you do not provide a drive, DOS uses the current one. Once this command is completed successfully, the MIRROR command displays a message indicating so.

After the MIRROR command is completed, DOS creates a MIRROR.FIL file and a MIRROR.SAV file. You will not see the MIRROR.SAV file, since DOS hides it, as well as making it a read-only system file. If these files already exist, DOS renames the old MIRROR.FIL file to MIRROR.BAK. You can skip creating a backup file by including the /1 switch. Use this switch for floppies, since you will usually need the disk space the MIRROR.BAK file uses for your own files. On hard disks, you will want to omit this switch, since you will probably have room for both the recent and the backup copy. Having the backup copy allows you to rebuild the disk if the original MIRROR.FIL file ever becomes corrupted.

The best time to use the MIRROR command to make an extra copy of the file storage information is when you first start the computer system. In Chapter 15, you will learn more about the AUTOEXEC.BAT file that DOS executes every time you start your computer. The MIRROR command is a command you may want to include in this file. You will also want to use this command before you perform widespread changes to the disk, since the UNFORMAT command, which you can use to reconstruct your hard disk, only remembers the status of the disk at the time you performed the last MIRROR command.

Using MIRROR to Monitor Deleted Files To use the MIRROR command to monitor the files you delete, enter MIRROR /TC, replacing the C, if necessary, with the letter of another drive on which you want to monitor deletions. You can even monitor deletions on several drives by entering a command like MIRROR /TC /TD, which monitors the deletions on drives C and D.

The MIRROR command will also record the file allocation table and root directory of the current drive (unless you supply a different drive between MIRROR and

the /T switch). This use of the MIRROR command loads part of the MIRROR command permanently into DOS. This is called a TSR or terminate-and-stay-resident program. TSRs remain in the computer's memory until you either restart the system or explicitly unload them from memory. Some DOS commands are TSRs, as well as other programs that are provided separately. TSRs offer features that are available by the touch of a keystroke or that monitor your actions so they can provide you with a feature at a later time. The MIRROR TSR remains in memory until you restart your computer or enter MIRROR /U. DOS does not unload the MIRROR command when you use the /U switch if another TSR has been loaded. All TSRs work this way: To remove any TSR from memory, you must first remove all TSRs that have been loaded after the one you want to remove. Once you use the /U switch, any files you subsequently deleted can only be recovered using the UNDELETE command and the /DOS switch, which forces DOS to use the directory information, as described below, to recover the deleted file.

When the MIRROR TSR is loaded, DOS keeps track of the files you delete and where the parts of the file are loaded. DOS keeps track of as many files as it can fit into the deleted tracking file called PCTRACKR.DEL. DOS does not create this file for you until the first time you delete a file. This file's size is by default determined by the disk capacity of the disk that DOS is monitoring. The default file size and number of entries DOS will monitor are shown in Table 6-1.

Table 6-1: The Entries DOS Will Monitor

Disk Size	Size of the PCTRACKR.DEL File	Number of Files Tracked in the PCTRACKR.DEL File
360K	5K	25
720K	9K	50
1.2 MB	14K	75
1.44 MB	14K	75
20 MB	18K	101
32 MB	36K	202
> 32 MB	55K	303

You can change the size and number of files for which the MIRROR command monitors deletions. When you use the /T switch and a drive letter, you can follow it with a hyphen (-) and a number between 1 and 999, which represents the number

of deleted files for which the MIRROR command will store the information in the PCTRACKR.DEL file (DOS calculates the file size).

When the PCTRACKR.DEL file contains the information for as many deleted files as will fit into the file, DOS records the deleted file information, starting from the top and overwriting the information for the first files that were deleted. When you are using MIRROR to monitor several drives, DOS monitors each drive separately so that the PCTRACKR.DEL file only contains the information for the deleted files on the drive of the PCTRACKR.DEL file.

When you use MIRROR to track deleted files on floppy disks, you are setting mirror tracking for a specific drive rather than a disk. This means that every time you delete a file on a disk drive that the MIRROR command monitors, DOS creates a new PCTRACKR.DEL file for that disk. However, you can put that disk into a drive for which the MIRROR command is not tracking deletions, and then when you delete a file from the disk, the deletion information is not incorporated into the PCTRACKR.DEL file. Also, if you hear two beeps when you delete a file on a disk that does not yet have a PCTRACKR.DEL file, it is an indication that DOS was not able to create this file, usually because the disk is so full that DOS could not create the PCTRACKR.DEL file before it deleted the file you wanted to remove.

Since the MIRROR command with a /T switch is a TSR, you will want to load it at the beginning of your DOS session. Also, some TSRs do not work together, such as two TSRs that both use ALT-F10 to display their menus. When you use TSRs it is wise to load only the ones you want and check that each one works.

Using MIRROR to Track Disk Partition Information To use the MIRROR command to make a backup copy of the partition information, enter

```
MIRROR /PARTN
```

When you enter this command, DOS reads the hard disk partition information, which includes how it is divided up into partitions and logical drives. DOS prompts you for the drive letter to save the partition information or suggests A as a default. Either use this default or type another letter and press ENTER. The disk you use should be a disk other than the hard disk you are recording the partition information on, because if the partition information on the hard disk becomes damaged, you will not be able to get to the file containing the partition information if it is on the same disk. The partition information is in a file called PARTNSAV.FIL. Once you use this command, you should put the floppy in a safe location. Using the MIRROR command to record partition information is something you should only do when you change the partition information of the hard disk.

Undeleting Files

If you realize that you have accidentally deleted a file you need, you can use the UNDELETE command to restore it. The UNDELETE command can be used to restore one or more files that you have deleted, provided DOS has not used the space the files once contained to store other data.

You can undelete files by entering UNDELETE and either the name of the file to undelete or a path. If you enter only a path after UNDELETE, DOS assumes that you want to restore all of the deleted files in that directory. If you do not provide a filename or path information, DOS assumes that you want to restore all deleted files in the current directory.

If the directory has a delete tracking file created with the MIRROR command, when you enter this command, DOS displays the number of file entries in the PCTRACKR.DEL file and the number of these that can be recovered, cannot be recovered and can only be partially recovered. For each file that DOS can completely or partially recover, it displays the filename, size, date, time, attributes (described in Chapters 3 and 7) and when it was deleted. DOS then prompts whether you want to undelete the displayed file. If you type **Y**, DOS undeletes the file. If you type **N**, DOS does not undelete the file. Once you enter your response, DOS continues with the next file you may want to undelete. DOS does not prompt you for the files that it cannot recover.

If the disk does not have a delete tracking file, when you enter this command, DOS displays a message, then checks for a delete tracking file. Since a delete tracking file is not found, DOS checks the directory for deleted files and displays the number of deleted files and the number that may be recovered. For each deleted file, DOS displays the filename (with a question mark for the first character), its size, date, time and attributes (described in Chapters 3 and 7) and then prompts whether you want to undelete the displayed file. If you type **Y**, DOS prompts for the first character of the filename and substitutes the question mark with your entry. If you type **N**, DOS continues with the next file you may want to undelete.

The UNDELETE command has several switches. You can include /LIST to list the deleted files in the directory that DOS will be able to recover, without actually recovering any of them. This lets you review the files you may be able to restore without actually restoring them. The list includes the same file information displayed when you undelete a file. You can use the /DT switch to recover the deleted files that are listed in the delete tracking file. You can also include the /DOS switch so the UNDELETE command ignores the delete tracking file. Finally, you can use the /ALL switch to recover all deleted files without the confirmation process. With this switch, when the disk does not have a delete tracking file, DOS substitutes a # for the question mark that starts the filename. If a file already exists

with that filename, DOS uses one of the characters in this list: %, &, -, 1 through 9, or A through Z. You cannot combine the /DT, /DOS or /ALL switches, although you can combine any one of them with the /LIST switch.

The UNDELETE command has several limitations. First, it cannot undelete a file if another file has replaced it. Second, UNDELETE cannot recover a directory deleted with the RD command covered in Chapter 8. Nor can UNDELETE restore a file that was in a deleted directory.

Retrieving Files On a Damaged Disk

When a disk becomes damaged, you may not be able to use it. A disk can be damaged by mishandling, such as accidentally using it as a saucer for your coffee cup or putting it next to a magnet. When a disk is damaged, DOS displays messages indicating that it cannot read the disk or it has a bad sector. When this happens, you can use the RECOVER command. The RECOVER command reads a file sector by sector and combines the readable data into one file.

To recover one file, use RECOVER and the filename, including any path information if the file is not on the current disk. You will know that you need to recover one file versus an entire disk when several of the files are readable but one or more are not. An example of this command is:

```
RECOVER A:MY_FILE
```

This will read as much information from the file as it can and put this information into a file called FILE0001.REC in the root directory. Since a root directory can only store a limited number of files (for example, 112 on a 360K floppy), the RECOVER command will only recover files until the number of directory entries in the root directory are full. This may prevent you from recovering all of the files on the disk if the disk originally had more files than the maximum number of directory entries allowed.

To recover all of the files from a disk, use RECOVER and the drive letter, as in RECOVER A:. You will need to recover all of the files from a disk when the directory of the disk is unusable (using the DIR command displays DOS error messages). DOS will try reading every file on the disk and put the readable contents of each file into files called FILE####.REC in the root directory, where #### starts with 0001 and increments one each time a new file is used.

Once you use the RECOVER command, you will want to review the contents of each file to either provide the missing information or erase the file if it is unusable. Some programs cannot use a data file if a section of the file is missing. Also, executable files (with a .COM or .EXE extension) will not work correctly if part of

the instructions the program uses are missing. You will also want to transfer the files to another disk. This is more important when you are using the RECOVER command to recover an entire disk of files. When the RECOVER command finds bad sectors on the disk, it marks them as bad so DOS will not later write additional information to the same location.

- ***NOTE:*** If you spill coffee, soda, or other liquids on an important 5-1/4 inch floppy, you can try these steps to retrieve the information off of the disk: First, shake off what liquid will come off. Then calmly open the sides from the seams at the bottom, using your fingernails and pulling the seams away from the disk. Next, remove the diskette from its jacket and hold the center of the disk with two fingers (do not touch the disk) and pour rubbing alcohol over it to remove the spilled liquid. Next, take a new, unused diskette from its jacket. To do this, use sharp scissors to trim the edge of the diskette opposite the openings where the disk head reads the diskette. Slide the new diskette out and slide the diskette that you rinsed with rubbing alcohol in (if you press the sides of the jacket together, it will open enough to fit the diskette in). Put the diskette into the drive and use the COPY or XCOPY commands described in Chapter 7 to transfer its files to another disk. If that doesn't work, use the RECOVER command to recover any data you can.

Unformatting a Disk

Occasionally, you may realize that you have formatted a disk that you did not want to reformat. DOS 5 includes an UNFORMAT command to restore a formatted disk to its prior condition, assuming new files have not been added to the disk and that you created a copy of the system information with the MIRROR command prior to formatting. The FORMAT command also creates the MIRROR files, so you can use the UNFORMAT command to recover files that you erased with the FORMAT command. DOS 5.0 can do this because it has a different way of formatting disks than previous releases of DOS. In previous releases, when you formatted a floppy disk, DOS replaced every byte on the disk with a blank character. This ensured that every spot on the disk was empty. In contrast, DOS 5.0 only erases every byte on the disk if you use the /U switch with the FORMAT command. When you do not use this switch, DOS does not erase the files on the disk. Rather, it erases the file allocation table (the area where DOS keeps track of the location of blank spaces on the disk) and the root directory (the files and directories DOS has stored on the initial directory on the disk). This is the same information stored in the MIRROR files. After the format, since the file allocation table and the root directory are empty except for the hidden MIRROR files, DOS uses the entire disk as if it were blank. Any files that you write to the formatted

disk replace the existing information and DOS stores new information in the file allocation table and the root directory about the new files that are stored on the disk.

If you realize that you want the contents of the formatted disk back, you can use the UNFORMAT command. To use this command, type **UNFORMAT** and the letter of the drive to unformat, for example:

```
UNFORMAT A:
```

When you press ENTER, DOS prompts you to insert the disk to unformat and press ENTER. Next, DOS checks the disk for the mirror file. If this file is found, DOS displays the date and time that the mirror file was created and prompts whether you want to use this mirror file to rebuild the disk. You can type **Y** to rebuild the system (root directory) of your disk or type **N** to cancel the command.

If the UNFORMAT command cannot find the mirror file (or you use the /u switch to omit the mirror file), the UNFORMAT command will prompt if it can search your hard disk for the appropriate mirror file, if it can search the disk to unformat, and if it can recover any files or subdirectories it finds. For each prompt, type a **Y** to continue or an **N** to halt the UNFORMAT command.

When DOS unformats a disk, it uses the information stored in the mirror file to recreate the file allocation table and the root directory. Using a mirror file provides you the greatest chance of recovering your files. When DOS rebuilds a disk without a mirror file, DOS must read the disk and search for any file and subdirectory information. When you unformat a disk without using the mirror file, you may not be able to recover your files.

When you use the /l switch, DOS first prompts for you to type **Y** to start the rebuilding process without making any changes. After it finishes listing the subdirectories (using SUBDIR.# for the subdirectory names) and files, DOS prompts whether you want to make the changes permanent. If any of the files cannot be completely recovered, you are prompted whether you want to delete or truncate the incomplete file. Type a **D** to delete the file or an **I** to truncate it.

If you use the UNFORMAT command with a hard disk, you will probably boot from a floppy disk (unless you have multiple hard disks). If the hard disk requires a device driver (which is loaded with a DEVICE command in the CONFIG.SYS file), you will need the CONFIG.SYS file on the floppy disk to load the needed device driver.

Rebuilding a Disk Partition

Another use of the UNFORMAT command is to rebuild the partition table of a hard disk if the partition information stored on the hard disk becomes corrupted. When you enter UNFORMAT /PARTN, DOS prompts you to insert the floppy disk containing the PARTNSAV.FIL file that the MIRROR /PARTN command created and prompts you to enter the drive letter. After you insert the disk and type the drive letter, UNFORMAT checks drive parameters that are stored in the PARTNSAV.FIL file with the disk partition you are trying to rebuild. If they are not the same, the command stops (this ensures that you are using the same PARTNSAV.FIL file that was created for the particular hard disk). After UNFORMAT /PARTN rebuilds the partition, it prompts you to insert a DOS disk in drive A and press ENTER to reboot. This starts the computer with the revised partition information. From this point, you can use the REBUILD command with the other switches or none at all to rebuild directories and files.

You can also use UNFORMAT /PARTN /L to display the current drive's partition table. This lets you see the current partition information without making any changes. The table this command creates assumes that sectors contain 512 bytes (the default), so if sectors contain a different number of bytes, the number of megabytes in the table will not be correct.

Moving and Replacing Files on a Disk

One reason you will need to transfer files from one disk to another is to ensure that a disk contains all of the files that you want, as well as the most recent versions of files. For example, if you have a project's budget file on several machines, you want to know that each machine has the most recent copy of the file. You can use the REPLACE command to replace files on one disk with files from another.

To use the REPLACE command, enter REPLACE followed by the name of the file to replace and where you want the replacement put. An example of this command is:

```
REPLACE A:*.* C:\BUDGET
```

This command replaces files in the BUDGET directory on drive C (the target location) with files of the same name from drive A (the source files). This means that if drive A has files called BUDGET_1 and BUDGET_2 and the BUDGET directory on drive C contains BUDGET_1 and SALES_1, the REPLACE command will replace the C drive copy of BUDGET_1 with the version from drive A. The BUDGET_2 file on drive A is not copied and the SALES_1 file on drive C does not

change. While the command is processing the files, DOS lists the files it is replacing. When it is finished, DOS displays the number of files that are added and/or replaced or indicates that no files were added or replaced.

The REPLACE command has several switches you can use to change the files to be replaced. You can use the /A switch to add only the source files that do not exist in the target location. This switch cannot be combined with the /S or /U switch described below. As an example, you can enter:

```
REPLACE A:*.* C:\BUDGET /A
```

Using the same files listed above, this command adds the BUDGET_2 file without altering the BUDGET_1 or SALES_1 files on drive C.

Another option is the /U switch. This switch only replaces source files that exist in the target location if the date and time of the files in the target location is earlier than the date and time of the source files. If you use this switch with the first example of the REPLACE command, DOS compares the dates and times of the source BUDGET_1 file and the same file in the C:\BUDGET directory. If the file in the C:\BUDGET directory is earlier/later than the BUDGET_1 file on the disk in drive A, the file is replaced.

Another option, especially useful when you use *.* for the files to use with this command, is the /P switch. This switch prompts for every file the REPLACE command may want to replace. If you enter:

```
REPLACE A:*.* C:\BUDGET /P
```

DOS will prompt you to replace the BUDGET_1 file. You can type **Y** to make the replacement or **N** to skip replacing the file and prompt for the next file DOS may replace.

When a file exists in several locations in the source directory, and you want to replace all copies of the file, use the /S switch. For example, you could use the /S switch as shown here:

```
REPLACE A:*.* C: /S
```

In this example, DOS goes through all of the directories on drive C to find files with the same name as the files on the disk in drive A. If DOS finds the files in any directory, it makes the replacements.

Besides using a switch to check subdirectories, you can also use a switch to have this command apply to read-only files (this attribute, which prevents you

from changing the file, is covered in Chapter 3). When you use the /R switch, read-only files are replaced. If you omit this switch and try replacing a read-only file, the REPLACE command immediately halts. The REPLACE command will not work with hidden or system files, even when you use the /R switch.

The last switch is /W, which waits for you to insert a disk before starting the REPLACE command. Use this switch when you are using floppy disks, so you can check that you have put the correct floppy disks in the disk drives.

Renaming Disk Files

You can rename files from one valid DOS filename to another. This lets you provide meaningful names for your files or change them to more appropriate names. This also lets you correct spelling errors you may have made when you created a file. To rename a file, use the RENAME command or its abbreviation, REN. When you rename a file, you are not changing where DOS stores the file. Instead, you are changing the name DOS uses for the file. You can rename a file either in the current directory or in another one. To rename a file in the current directory, enter a command like this:

```
RENAME FILENAME.OLD FILENAME.NEW
```

This example renames a file called FILENAME.OLD to FILENAME.NEW. You must include any file extension the filename uses so DOS can distinguish between files like FILENAME and FILENAME.OLD.

Another option is to rename a file in another directory, as in:

```
RENAME A:\FILES\FILENAME.OLD FILENAME.NEW
```

Notice in this example that you did not have to supply the path for the new filename. In fact, if you include it, as in:

```
RENAME A:\FILES\FILENAME.OLD A:\FILES\FILENAME.NEW
```

DOS will not accept the command, displaying an error message instead. DOS knows the path information of the renamed file by the path provided for the old filename. You can also rename multiple files with one RENAME command, as in:

```
RENAME FILE*.* MINE*.*
```

This example renames all files that start with the letters FILE and have any extension (including none at all), changing the first four letters of each to MINE. When you rename files using the asterisk and question mark wildcards, it is important that you replace the same number of characters. If the replacement filename has too few characters, only the number of characters used in the new filename are actually replaced. An example of this kind of mistake is:

```
RENAME FILE*.* MY_*.*
```

If you use this command, a file named FILEDATA will become MY_EDATA. Another problem arises when the replacement filename has more characters than the original. When this happens, DOS replaces every character provided by the new filename. An example of this kind of mistake is:

```
RENAME MY_*.* FILE*.*
```

If you use this command, a file named MY_DATA will become FILEATA.

When you use the RENAME command, DOS will not let you rename a file if a file with the new name already exists on the directory. If you try this, you will get the message **Duplicate file name or file not found**.

- ***NOTE:*** Be careful about changing file extensions. Some programs use the file extensions to indicate the format of the data in the file. If you change the file extension, the program may not be able to read the file correctly, since it will assume the data is in a different format than it is actually stored in.

At a Glance

In this chapter, you learned many new DOS commands that manage the information you store on disks. You learned commands that prepare the disks to contain your files, and several commands for putting files on the disks. The tasks that you can accomplish with the techniques in this chapter are:

- **Prepare a disk to accept data—FORMAT**
- **Divide a hard disk into partitions—FDISK**
- **Copy files from one disk onto another disk—DISKCOPY**
- **Compare all files on two disks—DISKCOMP**
- **Compare one or more files—FC and COMP**

- **Add a label to a disk—LABEL**
- **Display a disk label—VOL**
- **Check the status of a disk—CHKDSK**
- **Check the status of memory—MEM**
- **Erase a file—ERASE or DEL**
- **Undelete a deleted file—UNDELETE**
- **Recover erased files—RECOVER**
- **Duplicate system information of a disk—MIRROR**
- **Recreate system information of a disk—REBUILD**
- **Update files from one disk to another—REPLACE**
- **Change the name assigned to a file—RENAME**

Chapter 7

Protecting Your Investment

The files stored on your disk represent an investment of time and money. Just as you take precautions to protect valuable papers, you will want to protect important files from damage. Your investment in creating files can be protected in many ways. There are several changes you can make to the status of your files. You have seen how status changes can prevent the accidental deletion or modification of files, but you can also hide these files from view. Another precaution you can take is to make a copy of important files and store them in a different location. DOS provides several different commands for making these copies.

What You'll Learn

After reading this chapter you will be able to:

- **Change the status of files to hide them from view**
- **Use the COPY command to make extra copies of your files**
- **Use XCOPY to add further options to the copy process**
- **Create a backup of one or more directories**
- **Restore lost or damaged files**

Changing the Status of Files

DOS has four attributes that can be set for any file. These attributes select whether the file is displayed or hidden, can be changed, is a system file, or is archived. The archived file attribute is used by other DOS commands to determine whether a file is selected for use with the command specified. You might use this attribute when you only want to copy the files that have changed since the last time you made a backup copy of the files.

The information for the attributes of each of the files is kept in the subdirectory or directory. This information is stored in a concise manner in what is called an *attribute bit*. DOS also stores the name, size, date last modified, and time last

modified for every file in the directory. You can change the attribute of one or more files by using the ATTRIB command.

Displaying the Current Attribute Setting

To display the current attributes of the files in the directory, type **ATTRIB** and press ENTER. This produces a display like the one in Figure 7-1. If you want to limit the files for which you display attributes, you can enter the path and filename after ATTRIB as in **ATTRIB *.EXE**. DOS displays an R to indicate that the file is read-only (cannot be altered), an A for an archive attribute, an S for a system file, and an H for a hidden file (one that does not appear when you use the DIR command).

Figure 7-1: Attribute Settings for the Files in the Current Directory

```
C:\>attrib
     SH        C:\IO.SYS
     SH        C:\MSDOS.SYS
  A            C:\COMMAND.COM
  A            C:\D1_1.PIX
  A            C:\AUTOEXEC.BAK
  A            C:\COPYMEMO.BAT
  A            C:\THIS.BAT
  A     R      C:\A.BAT
  A            C:\MENU.BAT
  A     R      C:\Z.BAT
  A            C:\COPYMEMO.TXT
  A            C:\LIST.BAT
  A            C:\CONFIG.SYS
  A            C:\SCREEN00.PCX
  A            C:\DIDLY.BAT
  A            C:\AUTOEXEC.OLD
  A            C:\TEST1.TXT
  A            C:\TEST2.TXT
  A            C:\TEST3.TXT
  A     R      C:\1.BAT
  A     R      C:\F.BAT
```

The R in column 2 indicates the file is read-only.

Changing the Attributes

To add an attribute, enter ATTRIB and +r, +a, +s, or +h followed by the name of the file to modify. You can include multiple attributes, as in ATTRIB +r +h MY_FILE.EXE. You can also use the question mark and asterisk wildcards to make the command change the attributes for all the files matching the wildcard pattern

you specified. To remove an attribute, follow the same steps only use a minus sign in place of the plus sign to indicate the attribute you want to remove.

Taking a Look at the Attribute Options

The system attribute is used to flag files DOS uses in the boot procedure. This attribute is seldom changed since it only applies to a few files which are usually marked as system files when you format the disk or load the operating system.

In Chapter 3 you saw the read-only attribute used as one method for protecting data on your disk from accidental deletion. The read-only attribute provides two advantages. It protects you from entering a command such as DEL *.* and later realizing you did not want to delete every file. Second, it prevents someone else from changing a file you do not want altered. When a file is read-only, commands like DEL *.* have no effect on it. You also cannot modify read-only files with other programs such as your word processor. You *can* use them as read-only executable programs since executing them does not require a change to the file. The only way to modify or delete a read-only file is to modify its attribute bit so it is no longer read-only.

The hidden attribute hides a file from your directory listing while making it available when you specifically ask for it. When a file is hidden, it does not appear in most file lists, such as the one produced by the DIR command. DOS includes a count of hidden files with the CHKDSK command and lists them in the file list when you include the /V switch with the CHKDSK command. DOS will also list the hidden files when you use the ATTRIB command by itself or when you use the /a or /a:h switches with the DIR command. You may want to hide program files and files you do not change. Program files can be hidden and they will execute properly. You can also hide many of the files used by the program, but you will want to check that the program works after you hide files. You may start by hiding all files used by a program and then unhide the ones the program cannot find. You can also hide files to prevent other people from manipulating them since they will not know of their location. Since the files are hidden, they are also omitted from commands such as DEL *.* (which makes hiding files or making them read-only useful if you use this command too frequently). When you remove all of the non-hidden files, the hidden files will prevent you from deleting the directory they are stored in.

The archive attribute indicates files that have not been backed up. Later you will learn how you can use the /a switch for the XCOPY command to only copy files that have their archive attribute on. By default, all files have the archive attribute. You can use this command to switch the archive attribute off with -a and later, when you want to copy modified files, you only have to copy files with the

archived attribute on. The BACKUP and RESTORE commands can also use the archive attribute.

Using All Directories with ATTRIB

You can request that the ATTRIB command operate on all subdirectories at once. As you will learn in Chapter 8, subdirectories are divisions of a disk which enable you to organize your files into groups. You can use a /s switch with ATTRIB to find a file when you are unsure of the directory where the file is located, when you want to list all files in all directories, or if you want to set the file attributes of files in several directories. As an example, suppose you saw a file once called CASTLE, but you cannot remember the directory or file extension. You can find out the directory containing this file by entering:

```
ATTRIB \CASTLE.* /s
```

This command searches every directory on the disk for a file called CASTLE with any extension. The results will show there is a file called CASTLE.BAS in the DOS subdirectory and it will also display any attributes. In addition, you can use the ATTRIB command to list files across directories, such as entering the following to list the files with a .BAT extension:

```
ATTRIB \*.BAT /s
```

This command produces a list like the one in Figure 7-2 on the next page. Finally, if you want to protect these batch files with a .BAT extension, you can make these files read-only by entering:

```
ATTRIB \*.BAT /r+ /s
```

You can repeat the same command using /r- to return the files to not being read-only. When you use the /s to change attributes, you should be careful that you are only including files you wish to change.

Copying Files with COPY

A simple method of protecting a file is to make a copy of it with the COPY command. DOS lets you copy files to new locations and to new destinations. You can even copy a file to the same directory using a different filename. Use the asterisk and question mark wildcards to copy multiple files at once. The COPY command has three switches to set how DOS copies files.

Figure 7-2: Attribute Listing for All .BAT Files

```
  A           C:\AUTOEXEC.BAT

C:\>attrib *.bat /s
  A           C:\DOS\DOSCMD.BAT
  A           C:\DOS\1.BAT
  A           C:\RUP1\DOSSHELL.BAT
  A           C:\INSET\SERIAL.BAT
  A           C:\INSET\TEST.BAT
  A           C:\INSET\1.BAT
  A           C:\ONLINE\ZXCVBNM.BAT
  A    R      C:\COPYMEMO.BAT
  A           C:\THIS.BAT
  A    R      C:\A.BAT
  A           C:\MENU.BAT
  A    R      C:\Z.BAT
  A           C:\LIST.BAT
  A           C:\DIDLY.BAT
  A    R      C:\1.BAT
  A    R      C:\F.BAT
  A           C:\COPYMEM2.BAT
  A           C:\CNGPRMPT.BAT
  A           C:\ENVIRON.BAT
```

**.BAT causes batch files in the current directory to be listed. /s extends this to all directories beneath the current directory.*

The COPY command has two major parts. It must know the file or files it is copying and where it is copying them from. The source file can be either in the current directory, which DOS assumes, or in another directory when you provide the path information. While this information can include directory names, it must also include the filename. When you are copying files to or from another drive, DOS remembers the directory which was current the last time the drive was active and uses that directory.

The destination of the COPY command can use all of DOS's defaults to provide information to indicate where DOS should copy the files to. By default, DOS assumes you want to copy the file to the same directory using the same filename. If the file you are copying is in the current directory and you do not supply a new filename, DOS displays the message **File cannot be copied onto itself** and will not copy the file. Therefore, if you are copying a file in the current directory, you must include some information about where the file should be copied. By including a drive or directory for the destination, DOS will copy the file using the same filename, but a different directory. By including a filename, DOS will copy the file using the provided filename in the current directory. You may also provide both so that DOS copies a file using the new filename and a different directory.

For example, suppose you have two files with a .TXT extension in the \FILES directory called FILE_1.TXT and FILE_2.TXT. You can enter this command to copy these files after making the FILES directory the current one:

```
COPY *.TXT A:
```

This copies the files to drive A using the same two filenames. When you copy the files, DOS will display a message saying two files are copied. The files will be in the root directory unless another directory was current the last time you used drive A. You can also copy the files and change filenames as you copy them by entering a command like this:

```
COPY *.TXT A:\*.TEX
```

The above string copies the files with new names (they will have TEX file extensions instead of TXT). These files will be in the root directory since the back-slash follows the drive letter. When you supply a filename that is the same as a directory name, DOS copies the file to the directory using the original name of the file that is copied. An example of this is if you have a directory on drive A called FILES and you entered this command:

```
COPY FILE_1.TXT A:FILES
```

This command copies the file FILE_1.TXT from the current drive to the FILES directory on drive A. If drive A does not have a directory called FILES, DOS uses FILES as the filename of the copied file. You can also use the COPY command to copy files into the current directory by entering a command like:

```
COPY A:\FILES.TXT
```

This command copies the file FILES.TXT from drive A to the current directory using the same filename in the current directory. You can also provide both the source file and the destination filename as in:

```
COPY A:\*.* B:
```

This copies all of the files in the root directory of the disk in drive A to the disk in drive B. You do not have to provide *.* after B: since DOS will automatically use the same filenames.

When you copy a file, the copy of the file is identical to the original. This includes the contents of the file, as well as file attributes, size, date, and time information. You can have DOS double check that it has copied files correctly by including the /v switch as in:

```
COPY \DIR1\*.TXT A: /v
```

This copies all of the files in the DIR1 subdirectory with a .TXT extension to drive A. As the files are copied, DOS compares the original and the copy. If there is a difference, DOS displays an error message. Using the /v switch makes the COPY command perform slower since it must compare files after it copies them.

When the copying is complete, DOS marks the end of the file. It decides it has found the file's end in one of two ways. One method is by using the number of bytes. DOS normally copies one file to its new location until it copies the number of bytes that matches the file size. This is the default that the COPY command uses. Another option is to use an end of file marker. This means that DOS copies a file until it reaches an end of file marker. To copy files this way, add /a after the name of the file you are copying. You can specify the default of copying the number of bytes specified by the file size by adding /b after the name of the file you are copying. You can have DOS add the end of file symbol by adding a /a after the name of the file to copy the original. If you want to prevent the copy from having the end of file symbol, add a /b after the name of the file to copy the original.

Using COPY to Join Files

Another use of the COPY command is to combine several files into one. As an example, you may have several text files that you want to combine into a single file. You can use this with text files you create with word processors or the System Editor. For example, with batch files that are described in Chapter 12, you may want to combine several batch files to execute them in one batch file. The COPY command can quickly combine multiple files. To join several files into one, you must join the files with a plus sign as in:

```
COPY FILE_1.TXT+FILE_2.TXT FILES.TXT
```

This command copies the contents of FILE_1.TXT and then the contents of FILE_2.TXT to the file FILES.TXT. FILES.TXT will not contain the end of file markers for the two files, but will have one at the end of the file since as a default, the COPY command uses the /a switch when you use the COPY command to append files. If COPY did not omit end of file markers, text editors would only be able to

edit the first file portion of the appended files because most text editors only read files until the first end of file marker. You can also use wildcards for the source file of the COPY command rather than use the plus sign as in COPY FILE_?.TXT FILES.TXT, which produces the same result as the COPY command above. One potential problem is using a command like COPY *.* ALL_FILE which will at some point try copying itself into itself. When this occurs, you will see a message such as **Content of destination lost before copy**. This problem also occurs when one of the source files (except for the first) is also used as the destination. You can use the same file for one of the source files and the destination files if the file is the first source file as in:

```
COPY FILES.TXT+FILE_1.TXT+FILE_2.TXT FILES.TXT
```

In this case, the contents of FILE_1.TXT and FILE_2.TXT are added to the end of FILES.TXT.

Learning the Advantages of XCOPY

Besides using the COPY command to duplicate files, you can also use the XCOPY command. This command is similar to the COPY command except that it has more options. Like the COPY command, you can use any valid filenames for the source and destination filenames.

The switches the XCOPY command provides lets you: only copy files with a date later than the date specified, use the archive attribute byte to select which files are copied, prompt whether you want to copy each file, create new subdirectories, verify the copies, and wait until you are ready to copy files. The /v switch is identical to the /v switch for the COPY command since it verifies the copies.

Copying Files Based on the Archive Attribute

The XCOPY command can be used with the archive attribute. (You learned how to change the archive attribute earlier with the ATTRIB command.) To copy all files that have the archive attribute to drive A, you would enter a command like:

```
XCOPY *.* A:\ /a
```

This only copies the files in the current directory that have the archive attribute, but it does not change the archive attribute. If you use /m in place of /a, not only will DOS copy the files with the archive attribute, but DOS will also turn this attribute off for the copied files. This means that after you copy the files, all files

will have the archive attribute turned off. After a while, you can see which files need to be copied by using this command again since only the files created and modified after the archive attributes were turned off will be copied.

Copying Files Based on Their Date Entry in the Directory

Every file has a date entry in its directory. You can compare this date against a date you specify in the XCOPY command to determine which files should be copied. Only the files that have dates after the specified date are copied. An example of using the /d switch to do this is:

```
XCOPY *.* A: /d:01-10-91
```

This copies files that have dates after January 10, 1991. Files with dates before this date are omitted from the copying process.

Making the Copy Determination for Individual Files

If you want to copy most, but not all the files in a directory, you might want to ask DOS to copy all the files and then request a prompt before completing the copy for each file. The /p switch prompts you for a confirmation before it actually copies the file. For example, if you want to copy an assortment of files from the current directory to drive A, you might enter:

```
XCOPY *.* A: /p
```

When you pressed ENTER, DOS would prompt for each file and ask whether you wanted to copy each file. You can type **Y** and press ENTER to copy the file and prompt for the next file to copy. Type **N** and press ENTER to omit copying the file and prompt for the next file to copy. This is quicker than entering COPY commands for the files you want to copy.

Pausing Before Copying

The /w switch tells DOS you want it to pause and wait for your direction before beginning the copy operation. Using /w causes DOS to display the message **Press any key to begin copying file(s)**. This switch is used in batch files as described in Chapter 12. The batch file that uses this switch performs some steps such as formatting a disk that the XCOPY command will use, but before XCOPY begins copying files, the /w switch displays a message and allows the user to determine when they are ready to proceed.

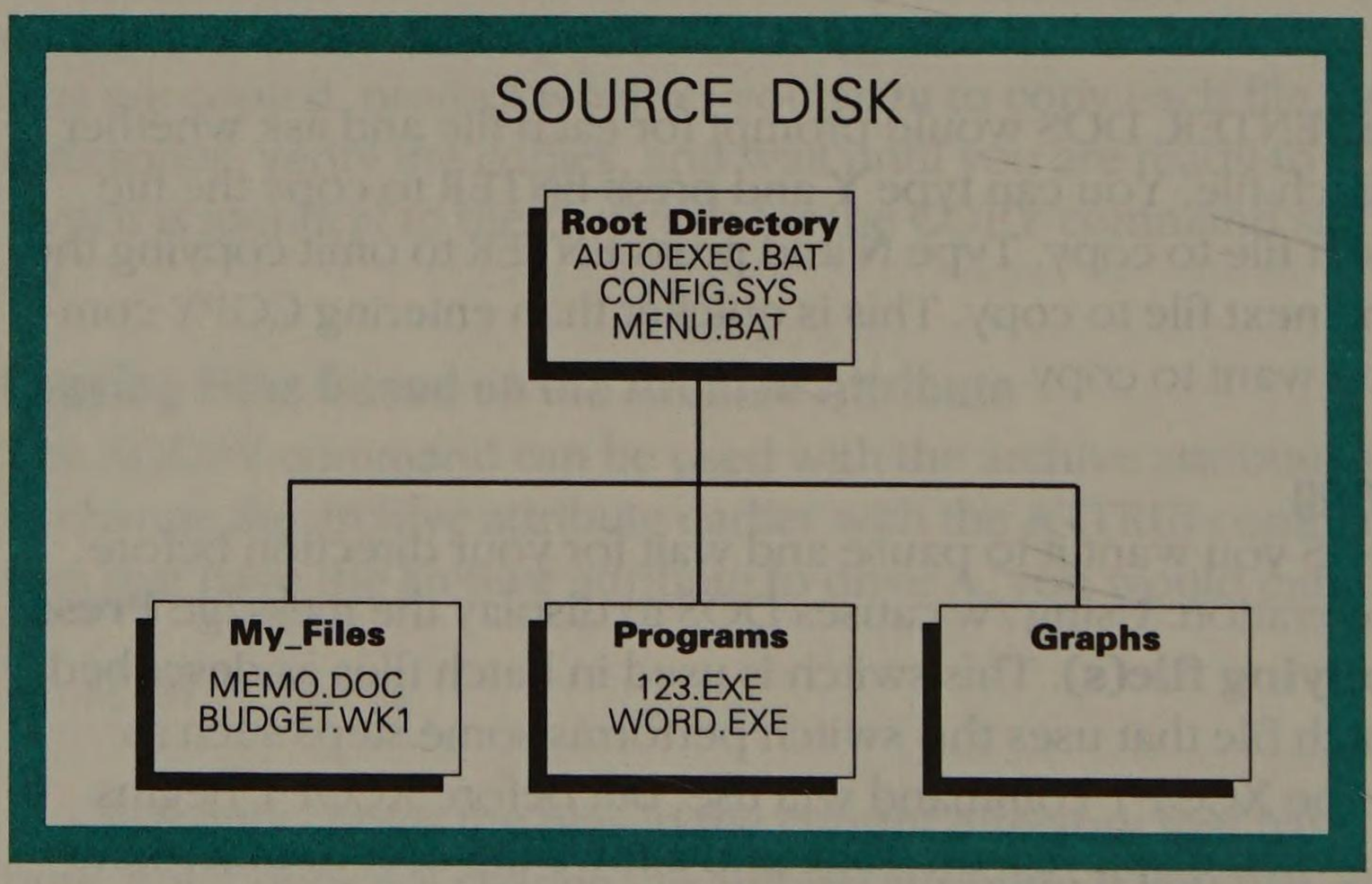
Using XCOPY with Subdirectories

The last two switches of the XCOPY command use subdirectories, which are described in the next chapter. When you use a command such as COPY A: B:, DOS only copies the files from the current directory on drive A to the current directory on drive B. If drive A has any subdirectories, the files in these subdirectories are not copied. On the other hand, if you have a disk organized like the one in Figure 7-3, you may want the destination file to include the subdirectories and the files in the subdirectories. To copy the files and the subdirectories shown in Figure 7-3 to another disk, you would enter a command like this:

```
XCOPY A: B: /s
```

This copies the subdirectories with files in them and the files that are in the root directory to the new location. When you use this command with the disk in Figure 7-3, you will have the results shown in Figure 7-4. You may notice that the GRAPHS directory is omitted in the new copy since the directory is empty. To create this empty directory you need to include the /e switch. This will include the GRAPHS directory in the disk shown on the left of Figure 7-4.

Figure 7-3: Disk Structure and Files to Copy

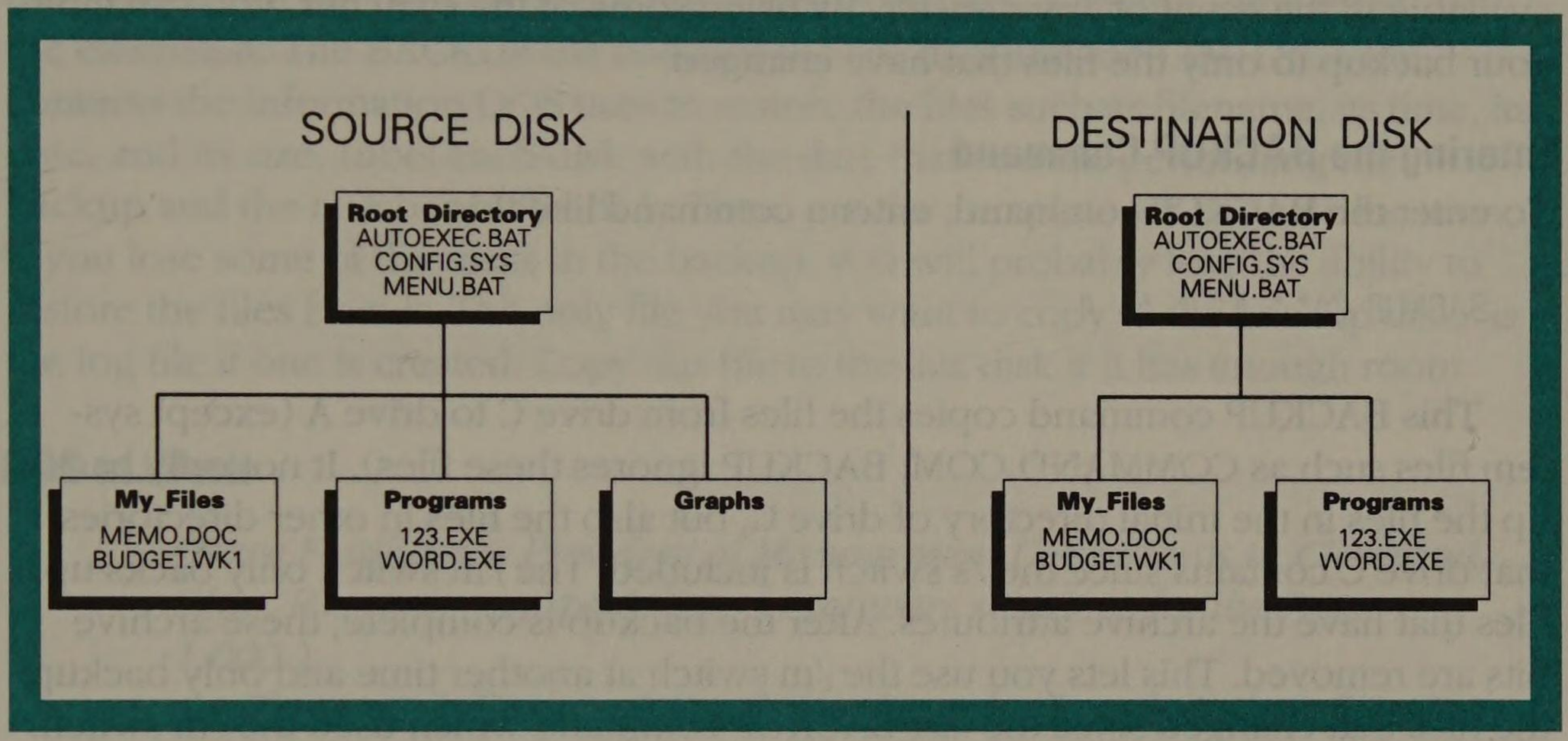


The root directory contains three files as well as three subdirectories. Two of the subdirectories also contain files.

■ ***NOTE:*** Use XCOPY to make entire disk copies of floppy disks since this command, unlike DISKCOPY, does not require that the two disks be the same format. Also,

any fragmentation (pieces of files that are stored in different areas instead of close together) can be reduced. The DISKCOPY command is best when you need exact duplicates of a disk, such as making a backup copy of software for archival purposes.

Figure 7-4: Results of Copying Disk Structure and Files



Creating a Backup

DOS includes a BACKUP command that makes backing up your hard disk or another disk easy. The BACKUP command is intended to create backup copies of files that may later need to be restored to the same disk. BACKUP works on hidden files and read-only files as well as the other files you may have on your disk. The BACKUP command can backup all of your files or just selected files. You can select files to be backed up based on the archive bit or on a date or time entry. You can also use the BACKUP command to format the disk as it copies—this makes backing up your files a one-command process.

One advantage of the BACKUP command is that it can handle large files and divide them into smaller files. This capability is useful when you create a file on your hard disk that is larger than the capacity of one of your removable disks. Without BACKUP you cannot make a copy of the file since neither COPY nor XCOPY can copy a file that will need to span two removable disks.

Realizing the Need for Backup Copies

When you backup a disk, you will want to put the backup disks in a safe location, preferably in a different building than the one where your computer is located. If anything happens to the original disks, you can use the RESTORE command described later to return the files to your disk. You should get into the habit of backing up your hard disk on a regular basis in order to have a recent backup available in the event of any disaster. By using some of the switches, you can limit your backup to only the files that have changed.

Entering the BACKUP Command

To enter the BACKUP command, enter a command like this:

```
BACKUP C:\*.* A: /s /m /l
```

This BACKUP command copies the files from drive C to drive A (except system files such as COMMAND.COM; BACKUP ignores these files). It not only backs up the files in the initial directory of drive C, but also the files in other directories that drive C contains since the /s switch is included. The /m switch only backs up files that have the archive attributes. After the backup is complete, these archive bits are removed. This lets you use the /m switch at another time and only backup the files that changed since the last BACKUP command which used the /m switch. The /l switch tells DOS that the BACKUP command should create a log of the files it is backing up. This log should include the date and time the backup is performed. Since a filename is not provided after the /l, DOS puts this information in a file called BACKUP.LOG on the current directory and drive. If a file with this name already exists, DOS adds the log information to the end of the file. If you wanted to provide a different name or select the location where the log file is created, you would enter a colon and filename after the /l, as in /l:C:\NEW_BACK.LOG.

A log file is needed when you want to check that the backup contains the files you need. This is a text file so you can use a word processor or the DOS TYPE command (see Chapter 2 in the section Looking at the Contents of Text Files) or the DOS PRINT command (see Chapter 9) to display the contents of the file.

Adding to an Existing Backup Disk

When you perform the BACKUP command, DOS prompts you for the first disk to use. Unless you use the /a switch described below, DOS will erase the contents of the disks you use. Therefore, you will want several disks that the BACKUP command can use to store the backup copies of files. You will see a warning that

all files that may exist on the disk will be erased. Since this occurs, be sure that you do not have any important information on these disks. When you press a key, DOS displays a message that it is backing up disk one and shows the files that are backed up to the disk. When the disk becomes full, DOS prompts for the next disk. This continues until DOS backs up every file. Each disk will contain a file called BACKUP with a three-digit file extension such as 001 or 002 to indicate the disk number of the backup, and a file called CONTROL with the same three-digit file extension. The BACKUP file contains the file data and the CONTROL file contains the information DOS uses to restore the files such as filename, its time, its date, and its size. Label each disk with the date that you are performing the backup and the number of the disk. Then, put the complete set in a safe location. If you lose some of the disks in the backup, you will probably lose the ability to restore the files from it. The only file you may want to copy to the backup disks is the log file if one is created. Copy this file to the last disk if it has enough room.

DOS at Work

(Dave Fordham is President of Management Consultants in Cleveland, Ohio. His company specializes in software solutions for the Fortune 1,000.)

The most difficult task is to have clients backup their data files. For most clients these data files represent a significant amount of work. A hard disk failure is unfortunately required to re-enforce the importance of regular backup.

A simple backup solution is to write a small batch file with the user's name as the first eight characters of the .BAT file. The procedure is as follows from the root directory:

```
COPY CON: username.BAT
CD C:\directory name consisting of the files to be backed up
ECHO OFF
CLS
ECHO  INSET THE BACKUP FLOPPY DISK INTO THE "A" DRIVE
PAUSE
COPY *.* a:
CLS
ECHO  CONGRATULATIONS username! YOU HAVE SUCCESSFULLY
ECHO  BACKED UP YOU DATA. PLEASE REMOVE BACKUP DISK NOW.
CD\
F6
```

Clients find it easy to type their own name to do the backup.

- ***NOTE:*** Do not use the backup disks for other purposes. When you use the backup disks for other purposes, you make the backups unavailable for restoration. When you no longer need a backup, you can use all of these backup disks for other purposes.

  The BACKUP command has other switches. The /a switch tells DOS to add the current backup to the end of an existing backup. When you use this switch, you must start the backup with the last disk in the set of backup disks from the prior BACKUP command. You can also use the /a switch when you want to start a set of backup disks with a disk that has information which you do not want to lose.

- ***NOTE:*** You cannot delete the backup files that DOS creates with the DEL command. You must reformat the backup disk with the FORMAT command if you no longer want to save the backup and would like to use the disk for another purpose.

Using Other Switches to Control the Backup

If the disks are not formatted before you use the BACKUP command, you will need to use the /f switch. When you use the /f switch, you must include a colon and the size of the disk that DOS will format as it backs up the files. The size of the disk is a number like 360, 360K, or 360KB for 360K disks (standard 5 1/4 inch diskettes) and 1440, 1440K, 1440KB, 1.44, 1.44M, or 1.44MB for 1.44 disks (high density 3 1/2 inch diskettes). This applies for 160K, 180K, 320K, 360K, 720K, 1.2 MB, 1.44 MB, and 2.88 MB diskettes.

The last two switches select files to backup based on the time and date of the files. When you enter /d: and a date, DOS only backs up files with dates after the date you specify. When you enter /t: and a time, DOS only backs up files with times after the time you specify. The format of the date and time must match the format of the date and time for the DATE and TIME commands.

- ***NOTE:*** Do not use BACKUP with redirected directories or drives that you have redirected with the ASSIGN, JOIN, or SUBST commands since you will only be able to restore them if you have the same redirected drives and directories when you restore the files. You will have better results if you remove the redirection and then backup the files.

Using RESTORE to Recover from Data Loss

After you have used the BACKUP command, you may need to use the RESTORE command to return files to the original disk. DOS cannot access the data in backup files with either COPY or XCOPY.

The RESTORE command will restore backups created with prior versions of the BACKUP command. Backups created under DOS 5.0 must be restored with the DOS 5.0 RESTORE command.

You cannot use RESTORE to restore system files such as COMMAND.COM and the two hidden system files. RESTORE can restore all of the files, a single file, or groups of files based on criteria you select with the switches the RESTORE command uses.

Entering the Restore Command

To use the RESTORE command, enter a command like this:

```
RESTORE A: C: /s
```

This command restores all of the files from the backup disks in drive A to drive C. Since the /s switch is included, DOS restores all of the subdirectories and the files stored within them. When you restore all the files, DOS starts with the first backup disk and continues restoring files from all of the backup disks. If you do not have all of them you will not be able to restore files from the first missing disk to the end of the backup set of diskettes.

If you only want to restore a single file, you might enter something like this:

```
RESTORE A: C:\PROGS\MY_FILE.EXE
```

This command restores the file MY_FILE.EXE in the PROGS subdirectory on drive C. DOS will start with the first backup disk and continue through all of them until it finds the ones it needs. This is different from the COPY or XCOPY command since you must enter multiple disks as DOS prompts for them. You can also use the RESTORE command to restore only the files in a selected directory as in:

```
RESTORE A: C:\PROGS\*.*
```

Using Other RESTORE Switches

The other switches limit the files that are restored. You can use the /p switch to have DOS prompt you when it is about to restore a file that is read-only or when the file changed since it was backed up. Figure 7-5 shows the warnings and

prompts that DOS displays when it encounters these two occurrences. The files that are not read-only and are the same date and time are restored. The opposite of this switch is the /m switch, which only restores files that are modified since the last backup; it does not display any prompts.

Another switch you may want to use, especially when you have accidentally deleted files, is /n. This switch only restores files that are not on the destination. As an example, if you accidentally delete some files in the PROGS directory, you can enter RESTORE A: C:\PROGS /n to restore these files from the backup.

Figure 7-5: Warnings and the Prompt the RESTORE Command Will Display

```
C:\>restore a: c: /p

Insert backup diskette 01 in drive A:
Press any key to continue . . .

*** Files were backed up 02-18-1991 ***

*** Restoring files from drive A: ***
Diskette: 01

WARNING! File COPYMEMO.BAT
is a read-only file
Replace the file (Y/N)?y
\COPYMEMO.BAT

WARNING! File COPYMEM2.BAT
was changed after it was backed up
Replace the file (Y/N)?
y
\COPYMEM2.BAT
\CNGPRMPT.BAT
\CMP.BAT
```

Warnings displayed for read-only file and when a file has changed since backup.

The last four switches use the file dates and times to restore files created before or after a specified date or time. You can use the /b: and a date to only restore files with dates on or before the date you specify. You can use the /a: and a date to restore only files with dates on or after the date you specify. You can use the /e: and a time to restore only files with times on or earlier than the time you specify. You can use the /l: and a time to restore only files with times on or later than the time you specify. The format of the date and time must match the format of the date and time for the DATE and TIME commands.

At a Glance

For most users, files represent a greater financial investment than their equipment. The hundreds of hours required to duplicate file entries is a valuable investment. In this chapter you learned some techniques to insure that your loss is minimal in the event of a system failure or hard disk crash. Some of the important techniques discussed in this chapter include the following tasks:

- **Hide a file—ATTRIB +h**
- **Display a hidden file ATTRIB—h**
- **Change other attriutes such as archive and read-only—+a, -a, +r, -r**
- **Copy a file—COPY**
- **Copy selected files—XCOPY**
- **Backup the current directory and all directories beneath it—BACKUP *.* A: /s**
- **Restore files from a backup disk—RESTORE A:*.***

Chapter 8

Organizing Your Hard Disk for Productivity

Hard disks have become such an important resource on computer systems that it is hard to believe that only a few years ago, hard disks were not part of personal computer systems. The hard disk is now used to store most software programs. This is especially important since today's full-featured database and word processing packages can require as many as 20 diskettes to store all their features. Without a hard disk, you would need to constantly change the disk in the drive to read different parts of the program. Also, as your use of the computer grows you will find that you have more and more data files. Although this data can be stored on diskettes, the hard disk offers a convenience since it is always there. You do not need to waste time looking for the correct disk in order to access your information.

What You'll Learn

The large volume of data that a hard disk holds can present a problem unless you organize the contents. The capacity of a hard disk is too large to treat it as one large storage pool. You will spend all your time reading through long lists of filenames unless you utilize DOS features that help you organize the disk. In this chapter you will learn to:

- Evaluate potential structures for your hard disk
- Select a structure for your disk
- Create, change, and remove directories
- Display the current directory structure
- Control the search path for files
- Use a different location for storing data
- Save time with a drive entry rather than a directory

Designing a Workable Structure for Your Data

When you first start to use your system, your hard disk has a root directory. A *root directory* is the main directory. Every disk including removable 5 1/4 or 3 1/2 inch diskettes has a root directory. On a removable disk you will probably store your data files in this directory. Since the capacity of a removable disk is itself a limiting factor, it is unlikely that you will exceed a reasonable capacity in the number of files written to a removable disk should you use this approach.

Your hard disk could have a capacity of 20 MB, 40 MB, or even more. With this amount of storage space you can store thousands of files. It is not practical to consider storing this number of files with only one directory listing. You would need to look through screen after screen of entries to locate the files you need.

DOS supports the use of directories to organize your hard disk. *Directories* are special files that contain entries for a group of files on your disk. Directories are not physical divisions of the hard disk, but make up a logical organization structure that can store information on the location of files—no matter where the files reside on your disk. Directories beneath the root directory are normally referred to as *subdirectories.* Although the discussion in this chapter focuses on a hard disk built into a PC unit, the same recommendations would apply to a disk used as a file server on a network and shared by a number of PC users or a floppy disk drive.

You can think of disk directories as similar to mini telephone directories that provide the names and addresses of individuals in a certain section of the city or business that provide services to other businesses. In large cities it has become impractical to publish one telephone directory. It would not be possible to bind the directory due to the number of pages that would be needed if the directory were to meet the needs of all telephone users. Although the situation and solution for disks are different than a telephone directory, conceptually the problem is quite similar.

There are physical limitations on the number of file entries that can be placed in the root directory but the practical limitations are normally reached first. Table 8-1 provides a list of different size disks and the maximum number of entries that can be placed in the root directory on any particular disk. Subdirectories beneath the root do not have these same restrictions, although common sense would indicate you should maintain the number of entries in each subdirectory at a manageable level.

■ ***NOTE:*** A good rule of thumb is to place no more than 60 entries in a directory. You can always create a new directory and copy files from your original directory to this new directory.

You can have as many as seven subdirectory levels beneath the root, for a total of eight different levels when you include the root directory. You will probably find that three or four directory levels are adequate for organizing your information. Exceeding this number of levels may add more overhead to locating files than using no subdirectories at all.

Files are automatically placed into the current directory when you write them unless you direct DOS to store the file in a different location. Likewise, DOS automatically searches the current directory when you request a file unless you direct DOS to look in a different location. For you to remember where files are located when you want to look at them again, it is important that you establish a logical structure. You might use an application-based structure, a departmental structure, or an individual structure. Anything that works for you is fine, although you will want to spend a little time considering what will be the optimal structure in light of any additional work you plan to perform with your computer in the future.

Establishing an Application-Based Structure

Application-based structures for hard disks are the most popular structures used today. With this structure a subdirectory is established under the root directory for each major application program that you use. A separate subdirectory is normally established for DOS to keep all of the DOS program files out of the root directory. When you install MS-DOS 5.0, the installation program copied many of the files to a DOS subdirectory which it created if one did not exist. If you are using dBASE, WordPerfect, and 1-2-3, the directories that you add beneath the root might look something like Figure 8-1. You will notice that each of the directories do not actually match the program names since directories have the same name limitations as data filenames and would not support WordPerfect as a directory name. Filename extensions are usually not used with directories but you can use a three character extension if you want to.

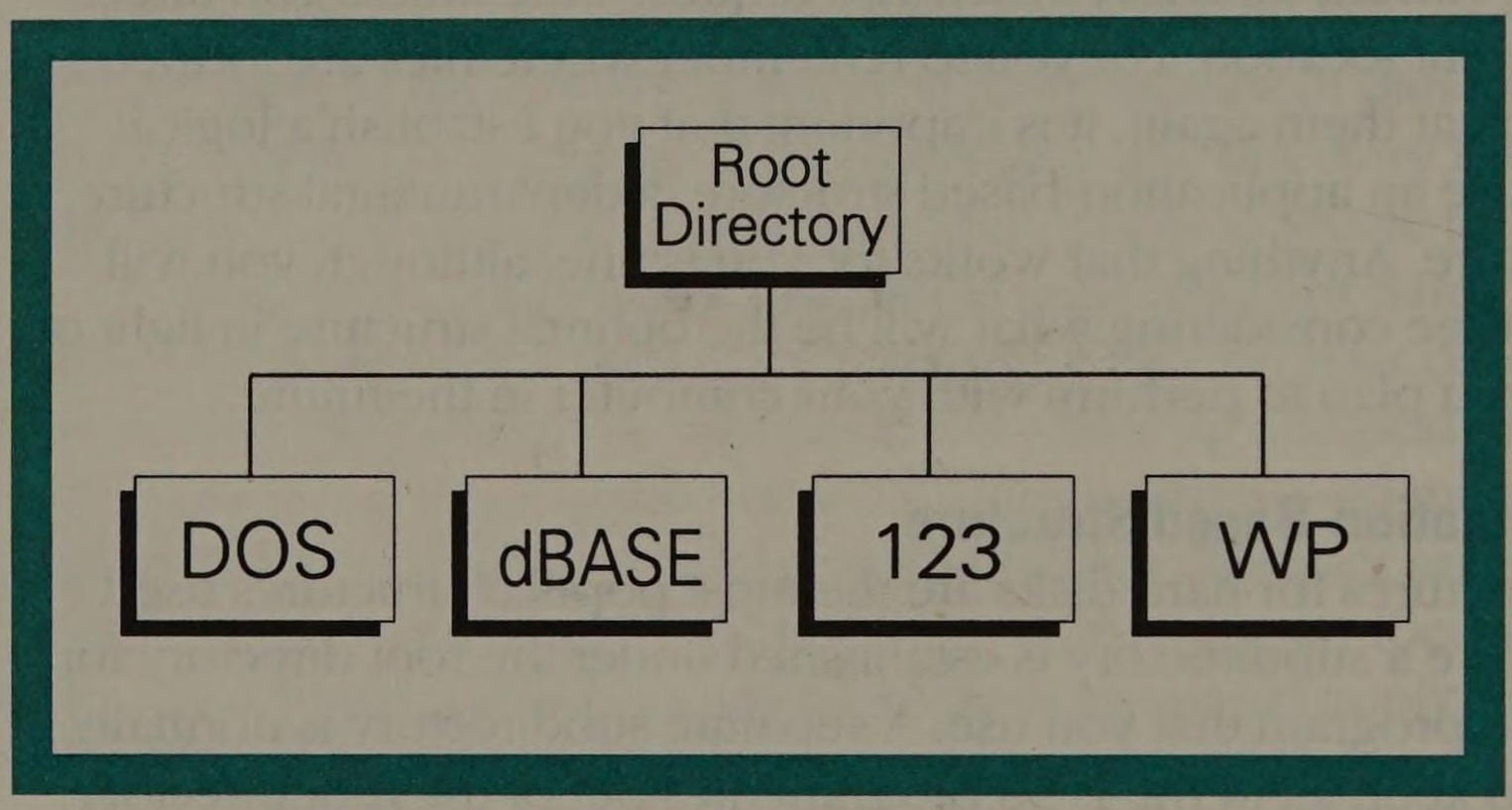
Establishing Lower Level Directories If you use the application approach for your directory structure you will want to establish at least one more level of subdirectories beneath the application level. The information for program files will be placed into the application level and the data will be stored separately. This approach helps prevent accidental loss of program files while you are attempting to delete data files and helps to control the number of entries in one directory.

Structure Remaining Directories Based on Work Tasks To determine the best structure for the remaining directory levels consider the types of work-related tasks that you perform with each of the applications and establish directories accordingly. If you

use WordPerfect for memos and reports you might consider these two directory levels:

\WP\MEMOS
\WP\REPORTS

Figure 8-1: Directories Added Beneath the Root Directory



Two levels of directories.

If you use 1-2-3 for accounting, sales projections, payroll, and budgets, you might consider these directories:

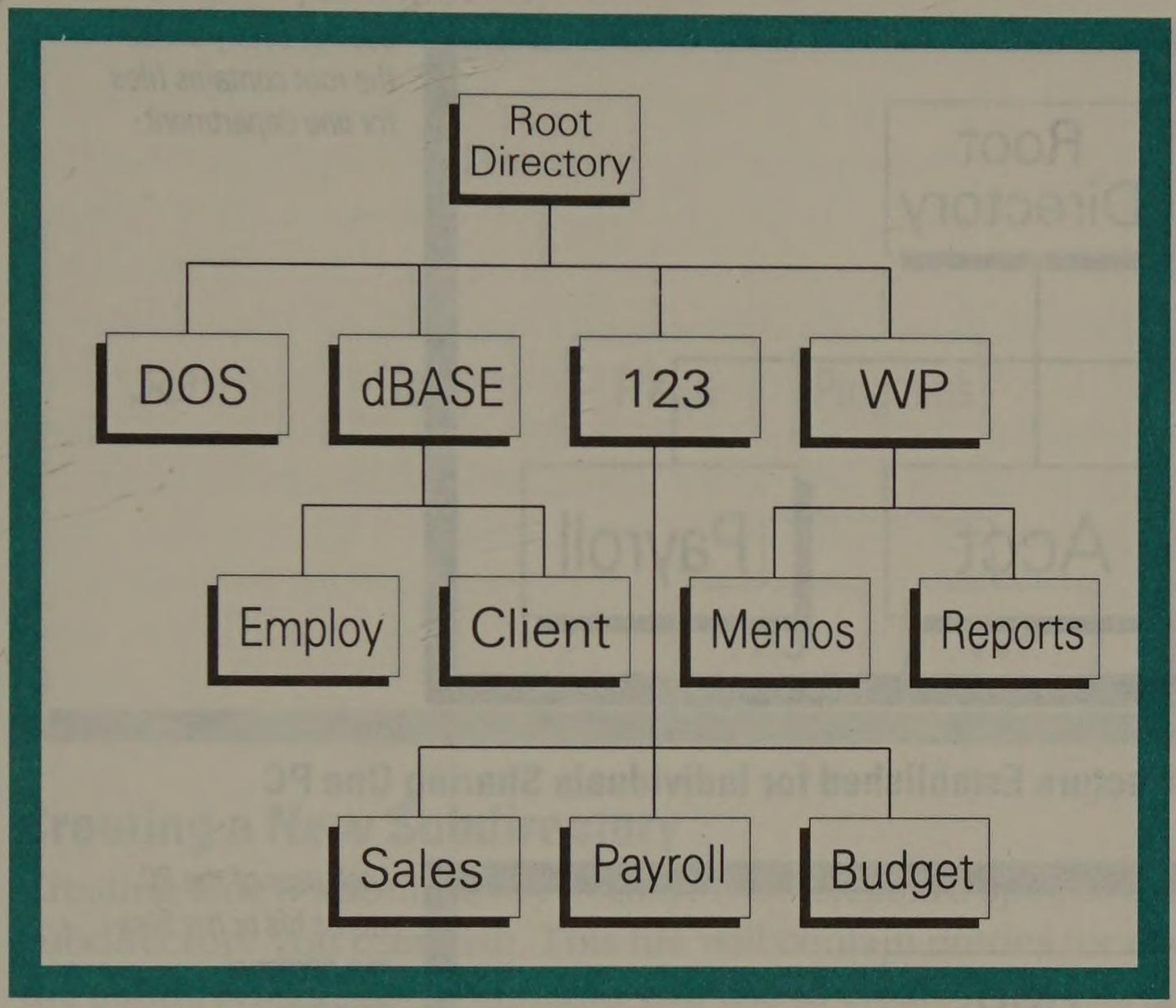
\123\ACCT
\123\SALES
\123\PAYROLL
\123\BUDGETS

If you use dBASE for an employee and a client database application you might add these directories:

\DBASE\EMPLOY
\DBASE\CLIENT

Your directory structure would look like Figure 8-2.

Figure 8-2: Second Level of Directories Added



Three levels of directories.

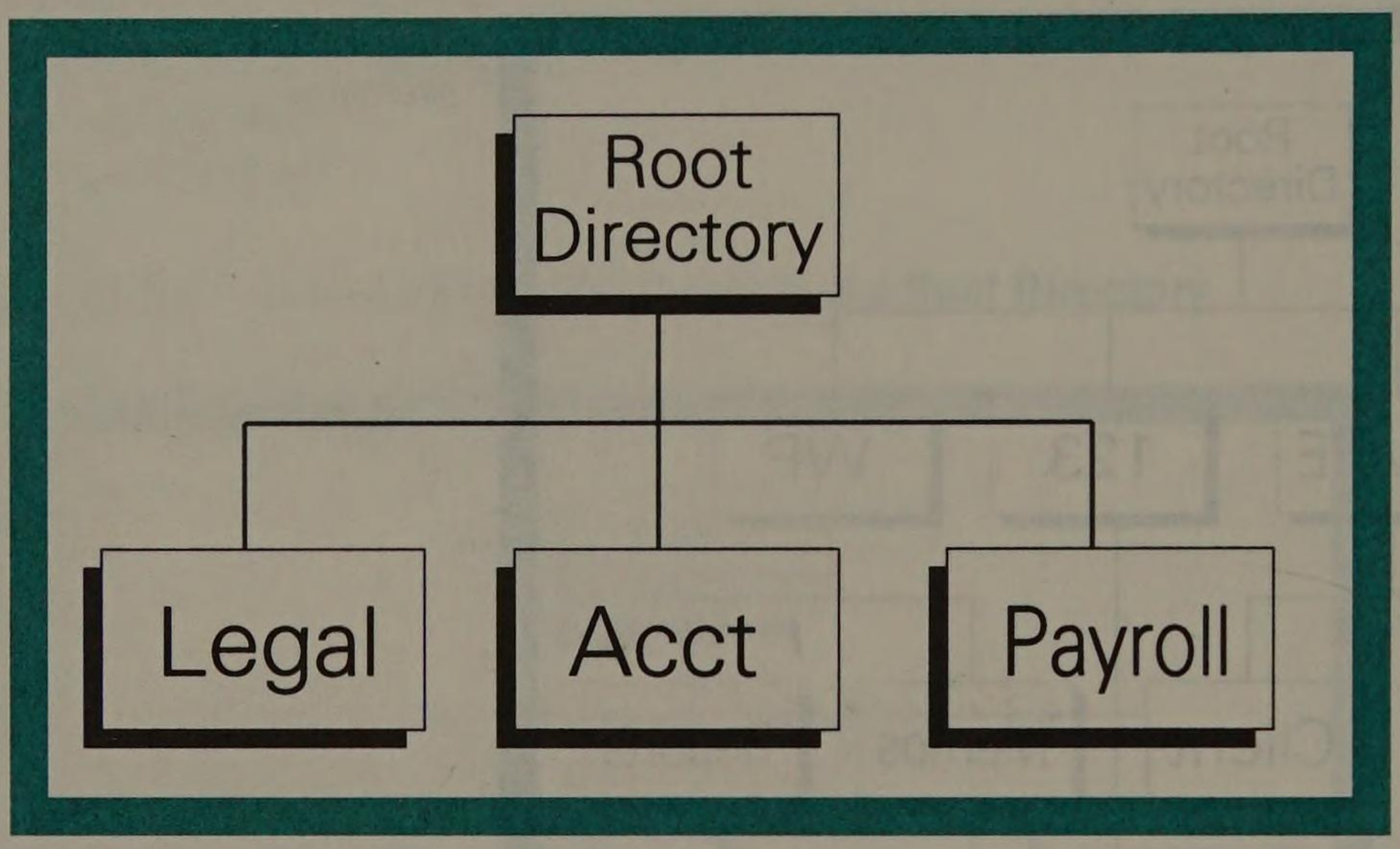
Establishing a Departmental Structure

When a PC is shared by two or more departments you might have the directories under the root directory indicate the department name. If the legal, accounting, and payroll departments share a hard disk you might have the directory structure shown in Figure 8-3. Since these departments have very different tasks and responsibilities it is likely that they use different software to perform these tasks. These applications could fill the level beneath the department entries. In situations where all departments use WordPerfect and 1-2-3, the application structure would be the best alternative. You would use the PATH command covered later in this chapter so you could start WordPerfect, 1-2-3, or other application from any of the departmental directories.

Establishing an Individual Based Structure

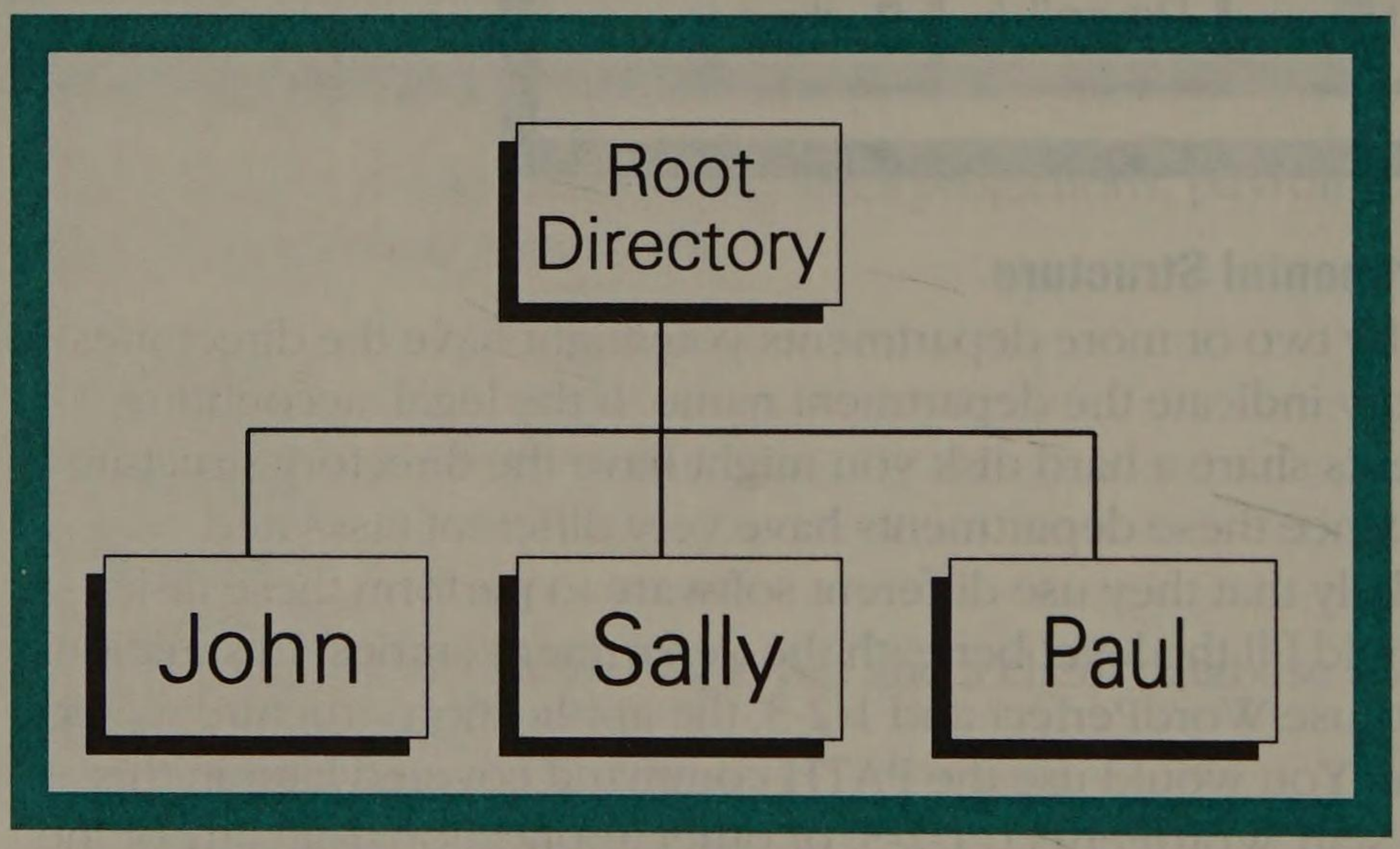
When several individuals share one computer system, you might want to use a directory structure that has individual's names as the second level of the structure. Figure 8-4 shows one possible structure.

Figure 8-3: Departmental Directory Structure



Each directory under the root contains files for one department.

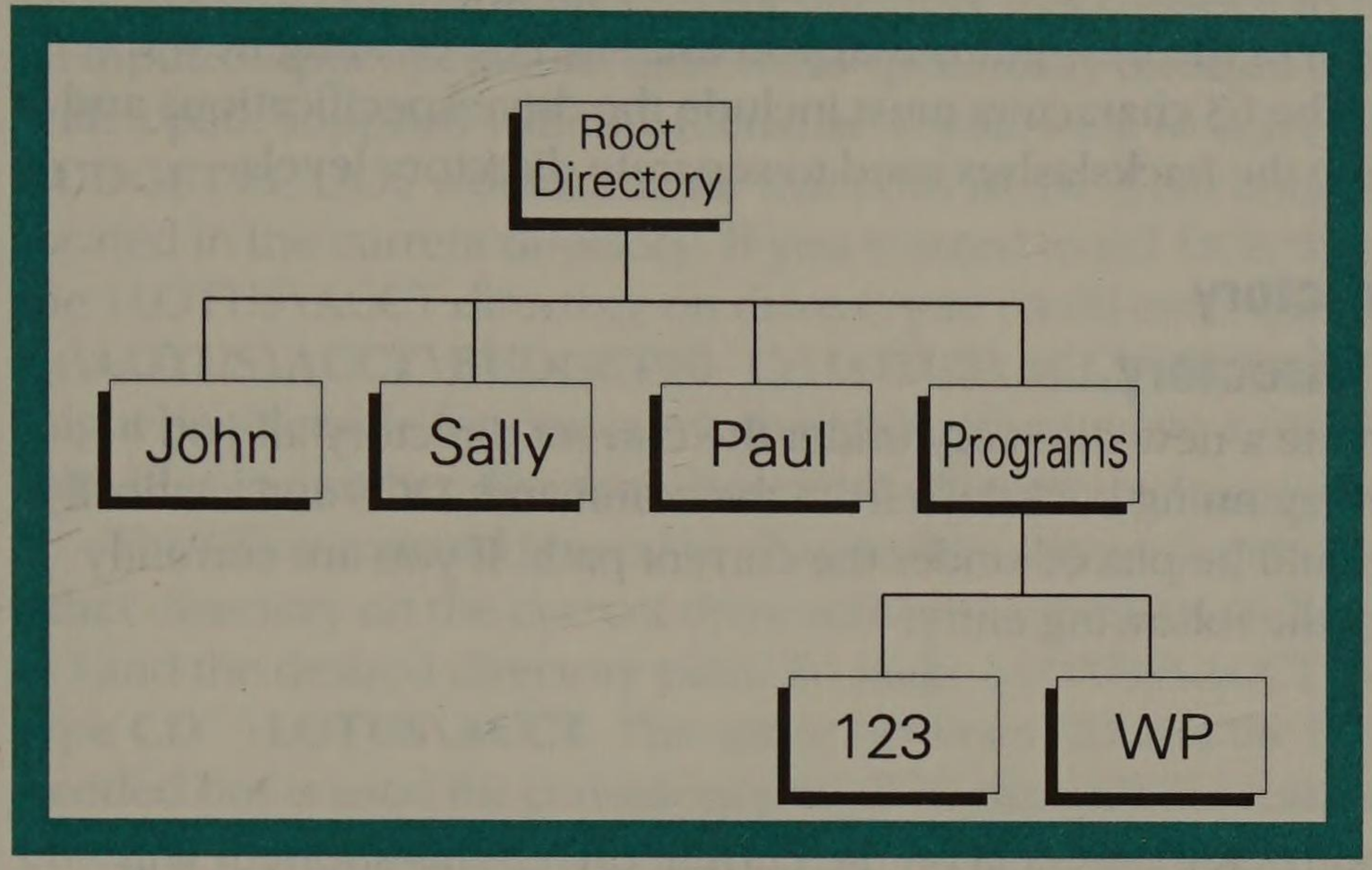
Figure 8-4: Directory Structure Established for Individuals Sharing One PC



Each user of the PC stores his or her files in a personal directory.

A user-based structure is most appropriate when the users work with different software packages. If all the users work with 1-2-3, you would not want to put a copy of 1-2-3 in each user's directory. It would be better to structure the disk with applications under the root directory and separate user directories under the application level. You could also keep the user's directories separate from the programs and have a structure like the one in Figure 8-5.

Figure 8-5: Combining a User and Program Structure



Programs are stored in 123 and WP with data files stored in John, Sally, or Paul.

Creating a New Subdirectory

Creating a new subdirectory is easy. DOS creates a special file for each subdirectory you establish. This file will contain entries for each file managed by the subdirectory. The command you use to create a new directory is the internal DOS command MD.

- ***NOTE:*** You can also enter this command as MKDIR although the extra three keystrokes that you will type do not change the command in any way.

Placing a New Directory Under the Root

To create a new directory under the root directory you use MD with a backslash as the first character in the directory path. Your entry might be:

```
MD \DBASE
```

This command creates a directory named DBASE under the root directory on the current disk. If you wish to create this directory under the root directory of another disk drive you must add the drive specifier to your entry as in:

```
MD B:\DBASE
```

This command establishes the directory DBASE under the root directory of drive B.

The length limitation of the new paths that you establish by creating new directories is 63 characters. The 63 characters must include the drive specifications and full pathname, including the backslashes used to separate directory levels.

Placing a New Directory Under the Current Directory

If you want DOS to create a new directory under the current directory all you need to do is omit the beginning backslash from the command. DOS automatically assumes your entry should be placed under the current path. If you are currently in the WP subdirectory the following entry:

```
MD MEMOS
```

creates the directory path \WP\MEMOS on the current drive. This shortcut will save you some typing time.

To create a subdirectory under MEMOS you can use the instructions in the next section or make \WP\MEMOS active by typing CD \WP\MEMOS. You can then add directories under MEMOS with entries like:

```
MD SMITH
MD JACOBS
```

which create C:\WP\MEMOS\SMITH and C:\WP\MEMOS\JACOBS.

Placing a New Directory at Any Location

You can add a new subdirectory at any location by first changing the current directory then entering all the new subdirectories as described in the preceding section. Unless you need to enter a large number of new directories under one existing directory, it may be easier to enter the pathname for each new directory rather than changing the current directory. All you need to do is specify the complete pathname for each subdirectory you wish to establish when you enter the MD command. If you want to create the subdirectory REPORTS under WP on drive C and the root directory of drive B is active, type the following:

```
MD C:\WP\REPORTS
```

Changing the Current Directory

DOS keeps a pointer in the current directory that causes it to use this directory for all input/output operations unless it is specifically directed to another location with a path supplied with the filename. If you were to enter **COPY BUDGET90 BUDGET91**, DOS would assume that both BUDGET90 and BUDGET91 were located in the current directory. If you wanted to tell DOS that both files were in the \LOTUS\ACCT directory on drive C you could enter the command as **COPY C:\LOTUS\ACCT\BUDGET90 C:\LOTUS\ACCT\BUDGET91**. Although this might be tolerable for one copy operation, if you have a series of tasks to perform with files in another directory, it is more efficient to change the current directory.

The CD command is used to change the current directory. You can make any other directory on the current drive active by typing CD followed by a backslash (\) and the desired directory path. To make \LOTUS\ACCT the active directory, type **CD \LOTUS\ACCT**. The space between CD and the backslash is not really needed but is used for consistency as all commands are followed by a space. After entering this command you could copy the two files in the previous example without providing a pathname since both the input and output file would use the current directory.

■ ***NOTE:*** You cannot change both the current drive and directory with CD. First change the drive by entering the drive letter and a colon and press Enter. Once you have activated the new drive, change the active directory with CD.

Each time you use a disk drive, MS-DOS maintains a pointer to the current directory on that drive. This means that if you change to the C:\123FILES directory and then change to drive D, when you use files from drive C as in the command DIR C:, MS-DOS assumes you want the files from the C:\123FILES directory. If you wanted to change the directory used in the other drive, you could use the CD command and include the full path information as in CD C:\WP, which changes the directory that will be used on drive C from 123FILES to WP. In this example, the space is required. The only time MS-DOS will change the directory it uses on another drive is with some terminate and stay resident programs (TSRs). TSRs are programs that are loaded and then disappear, but remain in memory so you can use the features of the TSR by pressing a special key combination. Since some TSRs use a selected directory, you may find that MS-DOS changes the pointer to that directory when it is not on the drive you are using.

Activating the Root Directory

You can activate the root or main directory by typing **CD **. Since the root directory is the only directory level without a name it is represented by the \. This entry always tells DOS that the root directory on the current disk should be active. If you are already in the root directory when you enter this command it has no effect.

Activating the Parent Directory

You can think of the directory above the current directory in the directory tree structure as the *parent directory*. Anytime you want to change from the current directory to its parent directory you can type

```
CD..
```

If you are in the directory one level beneath the root, this entry will have the same effect as CD \, but if you are in level 3 or lower of a directory structure, it will move the current directory up one level. For example, if the current directory is \LOTUS\ACCT, typing **CD ..** will make \LOTUS the active directory, whereas typing **CD ** will make the root directory active.

■ ***NOTE:*** If you are in the root directory you cannot proceed to a parent directory since you are already in the highest directory level. DOS will display the message **Invalid directory** if you attempt to move to the parent directory.

Removing a Directory

If you have directories that you no longer use, you should remove them from your disk. These unneeded directories waste space and distract from directories that contain important files. You will want to remove these directories. To remove a directory type

```
RD..
```

Using the Correct Syntax

The RD command is used to remove a directory. You can enter RD \ followed by the directory path to indicate which directory you wish to remove.

The entry RD \LOTUS\ACCT would remove the ACCT directory under LOTUS as long as this directory was empty and was not the current directory. The \LOTUS directory would not be affected by this command.

Observing Removal Limitations

You cannot remove a directory that contains entries for files. To remove a subdirectory with file entries you must first remove those files. You may wish to copy the files first or if you know they are not needed you can simply delete them. The other requirement for removing a directory is that the directory cannot be the active directory. Trying to remove the active directory would be similar to trying to dismantle the chair that you are sitting on. You can use the CD \ command to make the root directory active before attempting to remove a directory.

Looking at the Current Structure

The DOS Shell provides an easy way to observe a graphic representation of the disk structure. In Chapter 4 you saw how easy it was to change the active drive with a simple click of the mouse and get access to a graphic representation of any disk that is part of your system once you make the disk active.

You can also look at the disk structure from the DOS prompt. You will use the DOS TREE command to look at the organization structure. Although the display does not contain all the graphic elements presented with the Shell, the command can provide an overview or detailed look at the contents of a disk. To enter this command type **TREE** and press ENTER. This command lists the directories and subdirectories contained in the current one. You can select the directory and drive to list by entering this information after TREE as in **TREE C:** which lists the directory information for the entire drive as shown in Figure 8-6. Another option for this command is to add /F which lists the filenames in each directory (as shown in Figure 8-7) with the command TREE \DOS /F. The other option for the TREE command is to add /A which uses text characters instead of lines to draw the directory tree. This is important when you change the character set using code pages as described in Chapter 15.

Using a PATH Statement to Change the Search Path

When you make a request to execute a program or a batch file, DOS checks the current drive and directory for the program or batch file. Unless you tell DOS to look in other locations, a response of **Bad command or file name** will be displayed if DOS cannot find your file in the current directory.

You can use a PATH statement in either a batch file or at the DOS prompt to cause DOS to check other locations. You might even include your request in AUTOEXEC.BAT to have DOS make this change when you boot your system. Your PATH statement will cause DOS to look for files with a .COM, .EXE, or .BAT extension in these other locations.

Figure 8-6: TREE Listing Subdirectories in Drive C

```
C:\>tree c:\
Directory PATH listing for Volume DOS400
Volume Serial Number is 5C38-3850
C:\
├───DOS
├───RUP1
├───INSET
├───123R3
│   ├───WYSIWYG
│   └───ADDINS
├───WS5
│   └───MARY
├───CAPTURE
├───MICE
├───WP51
│   └───DOS
├───CSRTEST
└───HSG

C:\>
```

Tree listing shows only three directory levels.

Figure 8-7: Listing Files in the Directory with TREE

```
D:\>tree \wp_files /f
Directory PATH listing for Volume DOS400
Volume Serial Number is 5E3A-3A52
D:\WP_FILES
│   BOX
│   DOSOUTLN.BAK
│   DOSOUTLN
│   AGENDA2.BAK
│   AGENDA2
│   JEFF
│
└───MARY
        TUCKER
        THOMAS
        MARK
        KNIGHT
        MARTIN
        WAFER
        WILLIAMS
        PARKER
        MARTHA

D:\>
```

Tree listing shows both files and directories.

- ***NOTE:*** DOS only uses the directories in the PATH statement when you do not specify a path for the program you are executing. If you specifically direct DOS to look in another location in your entry, when it does not find the program or batch file in the location you specify, DOS will not look in the directory locations specified with the PATH statement.

- ***NOTE:*** It is common to see an equal sign (=) after some commands. PATH and PROMPT are two commands where it is used. The equal sign is treated as a blank space and can be used anytime you use a space in a DOS command.

The format of the PATH statement looks like PATH C:\DOS;D:\WP51. Optionally you can include an equal sign (=) after the PATH command. You will need to specify the drive and path for each directory that you want DOS to check with a limit of 127 characters on the full search path that you can check. Before you enter this command, you should consider which programs you want to be able to run from any directory and include the drive and directory information for each program in the PATH statement. As an example, applications like word processing or spreadsheet software can be used in multiple directories depending on how you store your data files for these applications. By including the directories for these applications in the PATH command, you can switch to the directory containing the data files you want to work with and load the software from that directory rather than the directory containing the software.

If you only want to tell DOS to look in the DOS directory for programs and batch files regardless of the current location you would type:

```
PATH=C:\DOS
```

- ***NOTE:*** Do not use unnecessary entries in your PATH statement. I have seen path statements that direct DOS to look in every directory on the hard disk. This wastes time searching directories where the file is unlikely to be found and can cause an accidental match with a filename in another directory when you misspell the name.

- ***NOTE:*** If your DOS prompt does not display the directory as well as the drive, add PROMPT=pg to your AUTOEXEC.BAT file. Customizing the DOS prompt is discussed in more detail in Chapter 15.

Special PATH Situations

There are two other ways that you can use the PATH statement. If you enter PATH with no parameters, DOS displays the current search path. If you enter PATH with only a ;, DOS confines the search to your current directory.

DOS always searches the current directory first for the program or batch file that matches your entry. If DOS does not find a match it checks the directories in the path statement in the order in which they are specified. The first matching entry stops the search.

Since it is possible to have a .COM, .EXE, and a .BAT file with the same name in a directory, you need to know the order in which they are executed. DOS checks first for a .COM file followed by an .EXE file, and finally a .BAT file. If a .COM file exists, the .EXE and .BAT files will never be accessed.

DOS at Work

(Johns Hopkins University offers non-credit business courses in their Downtown Center in Baltimore, Maryland. Room usage, purchase orders, and audio visual equipment are just a few of the resources that Yvonne D. Hassell, Facilities Coordinator for the John Hopkins Downtown Center must keep track of.)

I have a series of databases to track room usage, purchase orders, and the audio visual requests for our many classes at the Downtown Center" says Facilities Coordinator, Yvonne D. Hassell. "I need a series of batch files to keep my databases separate and easily retrievable. A copy of my batch file for accessing the audio visual database looks like this:

```
path = c:\dbase4;c:\dos;
cd \dbase4\av
dbase/t
cd \
path = c:\;c:\dos;c:\bat;
```

After setting the path for the dBASE directory, my batch file activates the audio visual database directory. I run dBASE IV with the command dBASE/t to avoid the dBASE IV Control Center menu and start dBASE quicker. After completing my dBASE session I go back to the root directory and reset my path to access other batch files.

When you enter a PATH command that includes the path information which MS-DOS should search, the directories selected with PATH override the previous PATH command. As an example, if you have the command PATH C:\DOS;C:\123 in the AUTOEXEC.BAT file, when you later enter the command PATH C:\WP51 at the command prompt, it replaces the PATH statement in the AUTOEXEC.BAT file.

This means that if you want to add another directory to the PATH command, you must enter all of the existing paths. The simplest method of doing this is to edit the AUTOEXEC.BAT file and reexecute it by rebooting the computer or just typing **AUTOEXEC** and pressing ENTER.

Using APPEND to Modify the Search Path

Just as the PATH command allows you to direct DOS to look in locations other than the current directory for program and batch files, APPEND performs the same function for data files. It is as if the directories you indicate with the append command are extensions of the current directory when you are searching for a data file since DOS does not stop looking for the data file until a matching file is located, or until all of the appended directories are checked.

Since DOS starts its search with the current directory and proceeds to the remaining directories in the order in which they are listed in the APPEND command, only the first matching data file name is returned. The APPEND command will also affect a DOS search for a program file since it checks the appended directories first after looking in the current directory and only when the program is not located in either of these locations are the directories specified by the PATH command checked.

Extending the Current Directory

APPEND is frequently useful when you work with files in several directories but do not want to constantly change the active directory. You can append several directories and DOS will treat them as if they are part of the current directory. This will speed your file retrieval process.

If the files you plan to work with are either in the current directory, (C:\DATA\REPORTS), B;\, or D:\SUMMARY\DATA, you can enter the following APPEND command to have DOS search all three locations for data files:

```
APPEND B:\;D:\SUMMARY\DATA
```

The APPEND command requires that you make the directory that you want to have the other directories appended to the current directory. After the APPEND command is entered, DOS will first check the current directory for a file and then check each of the subdirectories used in the APPEND command. When you append directories, the DIR command will only list the files for the current directory and will not include the files from the appended directory.

■ ***NOTE:*** Some word processing and database programs create a new file each time you use the program, retaining the original as a backup file. If the file you access is in a directory other than the current directory, the use of APPEND will not cause the new file to be written to that directory since new files will always be written to the current directory.

Using Special APPEND Settings

You can use APPEND without a parameter if you want to display the list of directories that are appended to the current directory. If you want to cancel the appended directories, entering **APPEND ;** will cancel the current settings.

There are several switches you can use with the APPEND command. Before you enter the APPEND command to append directories, you can use APPEND with /E, /X, and /path:on. When you use /E, DOS stores the directories selected with the APPEND command to the APPEND environment variable. In your own programs you can use this environment variable to test the directories DOS will check while looking for your files. This switch must be used in the first APPEND command you use after starting your system.

When you use /x, or the longer version /x:ON, DOS will check all of the directories selected with the APPEND command to find an executable file. If you enter /x:off, the default when you do not use /x:on, DOS will not check the directories selected with the APPEND command for executable files. When you use /path:on, DOS will look in the directories selected with the APPEND command even when you provide path information. This means that if you look for C:\DATA\REPORTS\MYFILE.BAT, DOS checks C:\DATA\REPORTS, B:, and D:\SUMMARY\DATA. If you use the parameter /path:off instead, DOS will only check C:\DATA\REPORTS because you provided the path for the filename. When you use the APPEND command subsequently and provide directories to search, you can use the /x:on, /x:off, /path:on, and /path:off although you cannot use /x and /e.

Using JOIN to Use a Directory Instead of a Drive

You might continue to use some older software designed in the days when a single drive PC was the norm. This software often used a single drive for both program and data files. With 360K disk capacities and a large program file, data files were limited in size. Some of this type of software would let you change the subdirectory but not change the disk used to work with your data. If you would like to continue to utilize this software, yet use other disk drives, consider using the JOIN command. The JOIN command will assign a temporary directory path to

replace another drive, usually a floppy disk drive. The program will write the data to the disk drive you select whenever the programs attempt to write data to the subdirectory.

Syntax of the JOIN Command

The format of the command requires you to enter JOIN followed by the drive that you want to treat as if it were another drive and directory. This is followed by the drive and directory that you want to use in place of the drive first specified. You might enter **JOIN B: C:\GRAPH**. Any directories or paths on drive B are treated as if they are in C:\GRAPH. When the DOS Shell is loaded, you will see this directory listed under a different drive with the files below it. In this example, when the Tree list shows the tree for drive C, you will see a directory called GRAPH. When you highlight GRAPH, the Files list will show the files on the disk in drive B. When you enter a command like TREE C:\, you will see the directory GRAPH. You can also use the command CD \GRAPH to change from the current directory on drive C to the B drive which due to the JOIN command is called the GRAPH subdirectory.

Limitations of the JOIN Command

There are several restrictions on the parameters used for the JOIN command. The first drive specified in the JOIN command cannot be used after it is used in JOIN. If you try, DOS will display a message telling you that you are using an invalid drive. Also, the path must be empty if it already exists. If it does not exist, DOS will create the new directory for you. This directory cannot be used for other DOS tasks while this command is in effect. If the directory exists and contains other files, DOS will display a message that the directory is not empty and the JOIN command will not take affect.

As mentioned earlier, the JOIN command is primarily used to use another location for the storage of data used with a program. Certain DOS commands will not accept the alternative location established with JOIN and for this reason its use is best confined to a batch file that invokes another program. After the program completes execution you can place JOIN B: /d in the batch file to cancel the effect of previous JOIN commands. This assumes that drive B is the drive you used in the original JOIN command and the drive you no longer want to refer to using the directory path. After you enter this command, you will have an empty \GRAPH directory on drive C. If you are uncertain of the current status you can enter JOIN with nothing after it to display the current status as in B: => C:\GRAPH. DOS commands that will not accept the location specified with JOIN include BACKUP,

CHKDSK, DISKCOMP, DISKCOPY, FDISK, FORMAT, LABEL, RECOVER, RESTORE, and SYS.

Using SUBST to Shortcut Long Pathnames

You can sometimes avoid the entry of many long pathnames by changing the current directory. This strategy only works when it is practical to change the current directory. When you cannot change the active directory, consider the use of SUBST to allow a substitution for a lengthy pathname. This command allows you to substitute an entry of E: for a path that otherwise might look in C:\LOTUS\ACCT. The command allows this substitution by establishing a virtual drive for the letter that you specify. This command is the inverse of JOIN since you are using a drive letter to represent a directory instead of using a directory to represent a drive.

- ***NOTE:*** You do not need a physical drive E because only a virtual drive is established. Since DOS only supports drive devices through E as the default you will need to add LASTDRIVE=letter to CONFIG.SYS when you want to use a letter greater than E (this is described in Chapter 15).

Entering SUBST Correctly

The syntax for the SUBST command can be SUBST with no parameters if you wish to display the current virtual drive assignments. More often the command is used to make an assignment and will appear like:

```
SUBST E: C:\LOTUS\ACCT
```

E: represents the virtual drive that will be used in place of the path specification that follows. The C: in the path specification that is to be assigned to E: is only needed when C: is not the current drive.

Using the Substitute Drive Letter

Once you have established the substitution letter for the path, you can use the drive letter anytime you want to access the path. Instead of entering DIR C:\LOTUS\ACCT you can enter DIR E:. The substitution drive cannot be used with BACKUP, CHKDSK, DISKCOMP, DISKCOPY, FDISK, FORMAT, LABEL, RECOVER, RESTORE, or SYS. When you are finished using the substitute drive letter, enter **SUBST E: /D** to stop using E for the directory. While the substitution is in effect, you can use either the drive letter or the path, unlike JOIN where you could only use the temporary directory assignment.

DOS at Work

(Judy and Joe Adamski are involved in writing Systems Analysis and Design, by Gary B. Shelly, Thomas J. Cashman, and the Adamskis.)

We share in the creative effort when we write books together. We each have our own computer and pass diskettes back and forth as we revise each chapter. We make extensive use of directories and sub-directories on our hard disks. We have found a combination of the COPY and SUBST commands to be most helpful when we transfer files between computers. For example, we use this SUBST command in our AUTOEXEC.BAT files:

```
SUBST H: C:\SOFTWARE\WORDPROC\BOOK1
```

Whenever we have to transfer revised files to our harddisks we simply issue the DOS command:

```
COPY A:*.* h:
```

and DOS transfers all files from the A: drive to the C:\SOFTWARE\ WORDPROC\BOOK1 subdirectory. We make a minimum number of keystrokes and errors. DOS allows us to concentrate our energies on the creative effort with a minimal amount of housekeeping details to distract our attention.

At a Glance

With the techniques covered in this chapter you should be armed with all the skills you need to establish an effective structure for the information on your hard disk. You can also use commands to direct DOS where to look for files and to look at the current disk structure. You have learned how to complete the following tasks:

- **Create a new directory —MD**
- **Remove a directory—RD**
- **Change directories—CD**
- **Look at the current disk structure—TREE**

- **Cause DOS to look in directories in addition to the current directory for programs and batch files—PATH**
- **Cause DOS to look in directories in addition to the current directory for data files—APPEND**
- **Use a disk drive instead of a directory—JOIN**
- **Substitute drive letter for a directory path—SUBST**

Chapter 9

Using Your Printer with DOS

In Chapter 4 you learned the basics of printing with the DOS Shell program. DOS has more printing abilities than just those in the Shell program. Most print capabilities are done with the PRINT command. The PRINT command prints text files you have stored on disk. You can also print the contents of your screen and your DOS session. Another change you may need to make while printing is changing the settings DOS uses for the parallel and serial ports that connect your computer to the printer.

What You'll Learn

In this chapter, you will learn how to:

- **Print text files**
- **Copy files to your printer**
- **Print the contents of your screen**
- **Change the settings DOS uses for the connection between the computer and the printer**

Using the PRINT Command

Most of the printing you do through DOS will be with the PRINT command. The PRINT command prints the contents of your file using *background printing*. This means that the resources of your computer are not dedicated to the print task. You can work on other tasks while DOS is printing your files. When you print information from another program you cannot exit the program without completing or canceling the print output requests that you have made. Because this output can be written to a file more quickly than you can send it to the printer, you can exit the program after writing the file and work with another program while DOS prints the file from the background.

Loading the PRINT Command

When you first load the PRINT command, the DOS print program is placed into memory. The Print program remains in memory making less memory available for other applications. If you are working with a system with limited memory, you will not be able to gain the benefits from background printing if the program you want to work with next requires all of your system memory.

The PRINT command includes many switches and parameters that you can use to alter how the PRINT command operates and the files it prints. When you use the PRINT command to print a file, you are adding the file to a queue that PRINT uses to control the order of the tasks it is printing. When the PRINT command is printing, SHIFT-Print Screen and CTRL-Print Screen will not work, nor can other applications print until DOS is finished printing the files in the queue. While PRINT is printing a file, you should not modify the file until PRINT is finished using it.

Most of the time, you will use this command by entering:

```
PRINT filename
```

Filename is the name of the file you want to print. The file referenced by filename should contain text or text with printer codes that is sometimes referred to as an *encoded file*. An encoded file contains specific directions for your printer that allow it to skip over the bottom page margin, use underline, and highlight text in boldface where requested. When you use this command, if your output does not start printing, you will want to check the list in Table 9-1, which lists the most common printing problems.

Making an Extra Entry the First Time You Use PRINT

The first time you use PRINT in a DOS session, DOS will prompt you for the output device that you wish to use for the output from this command. DOS will not prompt you if you have used one of the optional command switches to tell it the location. If you want to use the default device, PRN, all you need to do is press ENTER. You can also supply the name of a device attached to a different printer port. You will see more on the specific device options in the discussion of the /d switch that follows.

■ ***NOTE:*** Remember that the first time you use PRINT during a DOS session you must enter it at the dot prompt rather than selecting it from the DOSSHELL.

Table 9-1: Checklist for Solving a Print Problem

OK	Solutions
	Is the printer plugged in and turned on?
	Is there a cable between the back of the computer and the back of your printer?
	Is the On-line button turned on? Is the indicator lit?
	Is there paper in the printer?
	Is the cable between your system and the printer tight?
	Is the Form Feed light on indicating that print output is available, but the page has not been advanced?
	Does print Screen print the contents of the screen?

Put checks in first column

The PRINT command has six optional switches that you can only use the first time you use the PRINT command during your session on the computer. These switches are /d, /b, /u, /m, /s, and /q, and they are used in that order. These switches control the printer the PRINT command uses, the size of the buffer used by the command, the time the PRINT command has, and the number of files the PRINT command can store in its queue.

Selecting a Print Device The /d switch selects the printer the PRINT command uses. After /d: you must enter a print device name which can be PRN, LPT1, LPT2, LPT3, COM1, COM2, COM3, or COM4. PRN is the same as LPT1. LPT1, LPT2, and LPT3 refer to the parallel ports that are connected to your printer. Most printers are connected to LPT1. COM1, COM2, COM3, and COM4 refer to the four potential serial ports. If you do not enter /d: and a print device name, when you first enter the PRINT command, DOS displays a message like the one in Figure 9-1. If you want to use PRN, just press ENTER.

■ ***NOTE:*** In most system configurations, a single printer is attached to the first parallel port (PRN or LPT1) or the first serial port (COM1). If you are not sure of your configuration, try PRN as a print device before checking with your dealer regarding your configuration.

Figure 9-1: DOS Message When Print is Used the First Time

```
C:\DOS>print
Name of list device [PRN]:
```

Customizing the Size of the Print Buffer The /b switch customizes the buffer size that the PRINT command uses for storing information from the file it is printing. *Buffers* are work areas in memory that the system sets aside for storing file information so that disk access is not required for each batch of information sent to the printer. The default is 512 bytes, but you may want to increase this number with a fast printer to decrease the time spent waiting for disk access. You can change the size of the buffer by entering /b: and a number between 512 and 16384. When the buffer is larger, printing is performed faster, but you are taking away from the memory DOS has available for other applications.

Changing the Time Slices Available for PRINT When you are performing background printing, you may be working on another application in the *foreground*. The foreground application will be given most of the resources of the computer system and the background application will take longer to complete because it is sharing the system resources with the foreground application. *Time slicing* is a term (coined in the early days of computers) to refer to a similar situation on a larger computer where each application was given a short time to perform part of its work and then the next application was provided its slice of the total time. The computer keeps track of the order in which time slices should be given and how much time should be allocated to its application.

The next three switches affect the time allocation to PRINT and other programs running concurrently. DOS uses clock ticks which occur 18.2 times a second to keep track of time allocation or time slicing tasks. The /u: switch followed by a number between 1 and 255 sets the number of clock ticks that the PRINT command will wait until deciding the printer is too busy to accept more information. After the time has passed, DOS switches from the PRINT command to the other tasks you are performing even if the PRINT command still has more time. The /m: switch sets the number of clock ticks the PRINT command has to send information to the printer. The /s: switch selects the time slice or the number of clock ticks that other applications have until the PRINT command sends more information to the printer. The /m and /s switches allocate the time DOS spends on the PRINT command and the other DOS commands or applications you are using. These three switches are followed by a number between 1 and 255. The

defaults for the three switches are 1, 2, and 8. For an optimal entry, set /m to the time the computer needs to fill the printer's buffer (memory in the printer to temporarily store characters) and set /s to the time the printer needs to print the contents of its buffer.

Determining the Number of Files the Print Queue Can Hold The print queue keeps track of the print requests you have made that have not as yet printed. The final special switch you can use the first time you print is /q: followed by a number between 4 and 32. This switch sets the maximum number of files PRINT can hold in its queue. If you are printing many files, you may want to increase this number so the PRINT command queue has room for all of the filenames you are printing. The default is ten files.

Printing Data from Other Applications with PRINT

When you want to print data that you are using in other applications, you will normally use the printing features of the application. Most applications also support background printing and allow you to work on another task while printing a document. As long as you want to continue to work in this application you will not be motivated to use DOS to print documents from the package. When you decide to work on another application, you will not be able to print the original application's data from DOS unless it is stored in a text file or encoded print file which is a text file with printer directions for accessing the special features of the printer.

For example, word processing documents will often include special codes to determine how paragraphs are broken into lines, the justification the paragraph uses, and special attributes such as boldface. When you print the document with the word processor, the word processor prints the text as you want it to appear. If you use DOS to print the document, DOS prints the special codes as well as the characters so you will not have the results you want. On the other hand, if you print the output to a file instead of the printer, DOS will be able to print this file since the application has already added the directions for your printer. In these cases, you can use the PRINT command to print these output files since they will contain only the special codes the printer uses to print the desired result.

Adding and Removing Files from the Print Queue

The PRINT command lets you add and remove files from the print queue. When you use the PRINT command with a filename, the file is added to the print queue. Each time PRINT finishes printing a file, DOS removes the filename from the queue, advances the paper in the printer to the next page and prints the next file

in the print queue. When DOS adds a filename to the print queue, it adds the full drive and path information to the filename. This information for a single file cannot exceed 64 characters. You can use the asterisk and question mark wildcards to select the files to print. You can add files using the /p switch. This is the default when you use PRINT and a filename. This means you can print a file using PRINT MY_FILE, PRINT MY_FILE /p, or PRINT /p MY_FILE. You can also add multiple files to the queue by entering a command such as PRINT MY_FILE /p YOURFILE.

Each time you use a PRINT command, DOS displays a list of files currently in the print queue. If you only want the list of files in the print queue, enter PRINT with no parameters. Figure 9-2 shows a print queue list created this way.

Figure 9-2: Print Queue Listing

```
C:\>print


  C:\COPYMEM2.BAT is currently being printed
  C:\ENVIRON.BAT is in queue
  C:\BKUP.BAT is in queue
  C:\DOFILE.BAT is in queue

C:\>
```

There are three files in the queue that have not as yet started to print.

You can also remove files with the /c and /t switches. When you include the /c switch, filenames preceding and following the /c are removed from the print queue. If you cancel the file DOS is currently printing, DOS stops where it is printing, sounds any printer alarm, prints the message **File canceled by operator**, and advances the paper to the next page to print the next file. The other way you can remove files from the print queue is with the /t switch. This sounds any printer alarm, cancels the printing of the current file, removes all files from the print queue, and advances the paper in the printer to the top of the next page. You can also combine the /c and /p switches. When you use both switches in the same PRINT command, the file to the left of the /c or /p switch and any files to the right (except those belonging to another switch) belong to the switch. For example, you could have this PRINT command:

```
PRINT A.DAT /p B.DAT C.DAT D.DAT /c E.DAT F.DAT
```

This PRINT command will print A.DAT, B.DAT, C.DAT, and will cancel E.DAT and F.DAT.

Using the COPY Command to Print

Another method for printing files is to use the COPY command with a print device as the destination. For example, if you want to print A.DAT, you could enter COPY A.DAT PRN. If you want to print to another print device, you can enter a different device in place of PRN. The disadvantage of the COPY command is that DOS does not print from the background so you must wait until the COPY command is complete. The advantage of using the COPY command is that DOS will not load the PRINT command which occupies some of the memory DOS uses for other applications.

Printing a Screen

You might have a directory listing or other data on your screen that you wish to have in a hard copy format. One of the simplest methods of printing information is printing the contents of your screen. MS-DOS makes this easy with the Print Screen key (which is labeled PrtSc on some keyboards). This key can be used to send text or graphics to the printer. Text and graphics are two different types of screens since the way the information is presented is different.

Using a Quick Key Sequence

The Print Screen key provides the quickest method of printing the contents of your screen. When you press SHIFT-Print Screen, DOS prints the contents of the screen to the printer. It prints a line at a time and you can often see a faint line as DOS scans the screen from the top to the bottom. Many of the extended characters (ones with ASCII values greater than 127) may appear different than they appear on screen. Any colors and attributes such as boldface do not appear. If your computer seems to lockup, it is because the computer cannot send the information to the printer and it is waiting for the printer to accept the information before it accepts more input from the keyboard.

If the display mode is CGA and your screen contains extended characters (such as the ones that draw lines), these characters will not print correctly. To make them print correctly, enter GRAFTABL and press ENTER. If you change code pages as described in Chapter 15 to get the proper extended character set that includes the symbols used on your screen, you may want to include the code page number after GRAFTABL. Also, you may want to use the /status or /sta switch to display the code page used by GRAFTABL. This program remains in memory and

uses part of the memory available for other applications. Once GRAFTABL is loaded, you can print screens that use extended characters with no problem.

Figure 9-3 shows a screen that contains extended characters. The lines that create the boxes will only print correctly with SHIFT-Print Screen when you load GRAFTABL. If you are using GRAFTABL in a batch file as described in Chapter 12, you may want to use the exit codes this command sets. This command sets the exit code to 0 when GRAFTABL is loaded the first time, 1 when GRAFTABL is loaded subsequently, 2 when a file error occurs, 3 when an incorrect parameter is provided that prevents the command from working, and 4 when the incorrect version of DOS is used.

Figure 9-3: Screen Displaying Characters from an Extended Character Set

```
C:\WP51\DOS>type menubox

                ABC Company
                MENU SYSTEM

          1. Spreadsheet Analysis (1-2-3)
          2. Word Processing (WordPerfect)
          3. Data Management (dBASE IV)

             Type a number for the
           application you want to use.

C:\WP51\DOS>
```

The box around the text is drawn using extended characters.

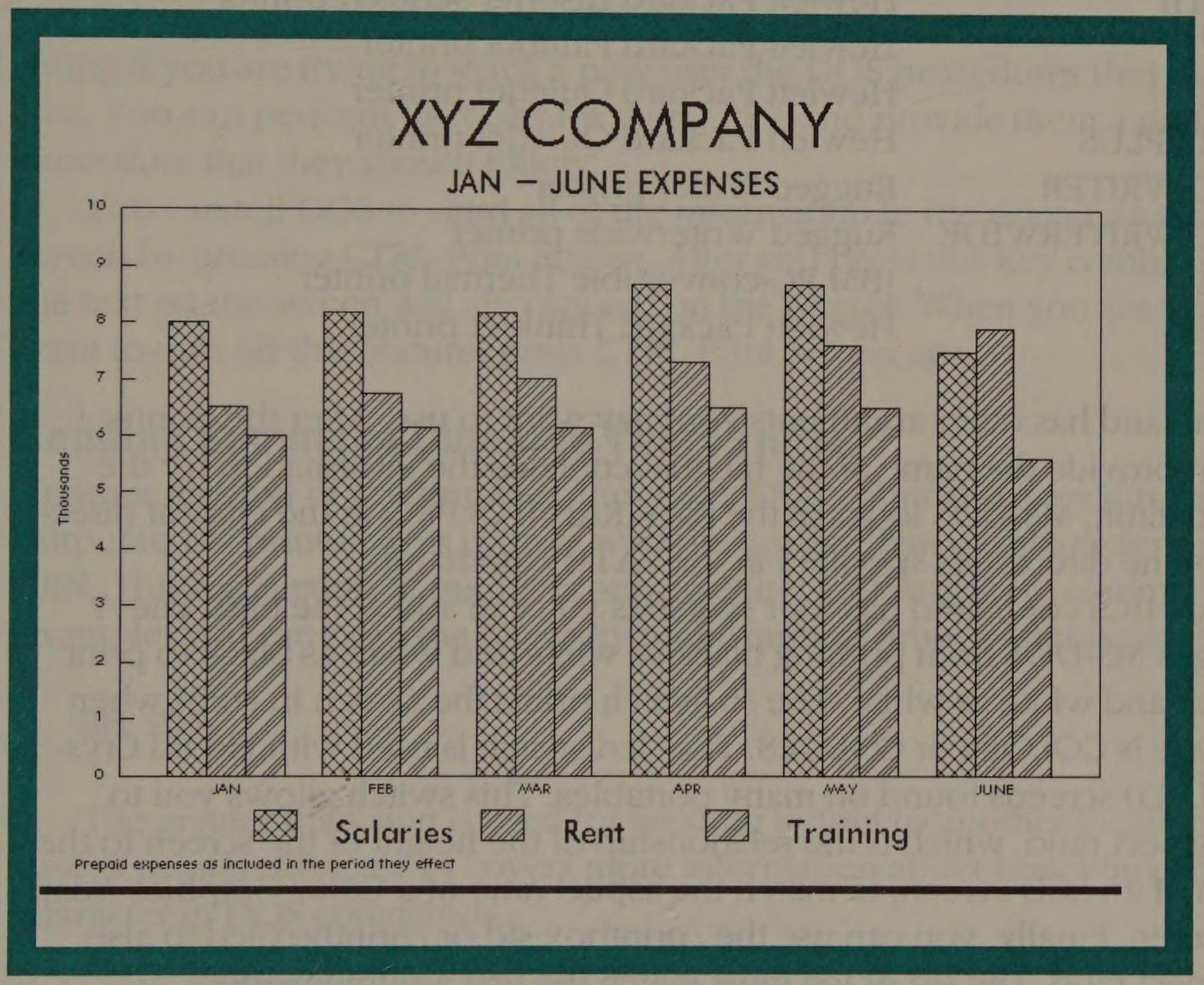
Understanding the Problem with Graphics Screens

While text screens are easy to print, graphics screens are not. This is because of the way DOS and other programs send information to the screen. In a text screen, DOS knows that the screen will contain as many as 256 different characters and each character is in an easily-defined row and column. When DOS prints a text screen, it reads one row of characters across the screen. Also, most printers can accept and print lines of characters without any special instructions. On the other hand, the graphics screen which is used for drawings as well as the DOS Shell program on EGA and VGA screens is not limited to text characters. Figure 9-4 shows a graphics screen. The contents of the screen are not limited to keyboard characters like you see on a text screen. Also, the potential locations for information is more varied. For example, an EGA screen can display 640 by 350 pixels across the screen. You can think of pixels as dots that create a grid on the

screen. As a further problem, printers do not accept graphic information as readily. If you try using Print Screen, you will not get the results you expect unless you follow the directions in the section on the GRAPHICS command that follows.

■ ***NOTE:*** The number of pixels varies greatly depending on the type of graphics adapter and monitor you use. A CGA adapter will display a grid of either 320 by 200 or 640 by 200 pixels. VGA adapters can display 640 by 480 pixels. The adapters which support a greater number of pixels offer improved image resolution on the screen.

Figure 9-4: Graphics Screen



Printing Graphic Screens MS-DOS includes a GRAPHICS command that allows you to print graphics screens using SHIFT-Print Screen. This command only needs to be used once during a DOS session. To use this command, enter a command like this:

```
GRAPHICS LASERJETII
```

This entry tells DOS to print graphics screen for a Hewlett Packard LaserJet Series II printer (or compatible). The printer types you can use and the printers they represent are:

COLOR1	IBM Personal Computer Color Printer—black only
COLOR4	IBM Personal Computer Color Printer with RGB ribbon
COLOR8	IBM Personal Computer Color Printer with CMY ribbon
DEFAULT	Any Hewlett Packard PCC printer
DESKJET	Hewlett Packard DeskJet printer
GRAPHICS	IBM Personal Graphics, Proprinter, or Quietwriter
GRAPHICSWIDE	IBM Personal Graphics Printer with a wide carriage
LASERJETII	Hewlett Packard LaserJet Series II printer
PAINTJET	Hewlett Packard PaintJet printer
QUIETJET	Hewlett Packard QuietJet printer
QUIETJET PLUS	Hewlett Packard QuietJet printer
RUGGED WRITER	Rugged Writer printer
RUGGED WRITERWIDE	Rugged Writerwide printer
THERMAL	IBM PC-convertible Thermal printer
THINKJET	Hewlett Packard ThinkJet printer

This command has other arguments you may want to use. After the printer type, you can provide the name of the file that contains the information for the printers. By default, MS-DOS looks in the file GRAPHICS.PRO in the current directory or one of the directories specified in the PATH command.

The GRAPHICS command has four switches you can add at the end. The /r switch changes MS-DOS from printing black as white and white as black to print black as black and white as white. The /b switch prints the screen in color when the printer type is COLOR4 or COLOR8. The /lcd switch is used with Liquid Crystal Display (LCD) screens found on many portables. This switch allows you to change the aspect ratio, which is the relationship of the height of the screen to the screen width of an LCD screen, to match the aspect ratio of a Color Graphics Adaptor (CGA) screen. Finally, you can use the /printbox:std or /printbox:lcd to also select the aspect ratio. The std or lcd must match the first argument after PRINTBOX in the GRAPHICS.PRO file for the printer and display mode you are using.

Once you enter the GRAPHICS command you can print a graphics screen by pressing SHIFT-Print Screen. Since graphics screens contain more information than text screens and DOS must add the printer information that the printer needs to print the screen correctly, printing a graphics screen will take several minutes.

While DOS is processing the screen, your system will behave as if it is locked up. Loading the GRAPHICS command causes the use of RAM memory.

- ***NOTE:*** To reclaim the memory used by the GRAPHICS command you can reboot your system when you do not need to print graphics screens any more. Since GRAPHICS only uses 6K of memory you will not want to take the time to do this unless you are really tight on memory.

Printing Your DOS Session

If you want to print a record of all your entries at the DOS prompt you can turn on the print feature for the entire session.

When you are first learning the DOS commands, you may want to see a printed copy of the commands you enter and their output. You may also want this listing if you are trying to teach a new user the DOS procedures they should follow. You can perform the tasks for them once and provide them a printout of the procedure that they should follow.

You can tell DOS to send all of the information to the printer as well as the screen by pressing CTRL-Print Screen. After you press this key combination, all of the text on the screen will also appear on the printer. When you are finished and want to turn off this feature, press CTRL-Print Screen again.

Sending Command Output to Your Printer

Another method of printing the output of a DOS command is by directing the output to the printer. This is done with the greater than (>) character followed by PRN. This redirection of output is only in effect for the current command. For example, you can print the directory of the current drive with this command:

```
DIR >PRN
```

The greater than sign is used for sending output to another location, in this case, a printer. Chapter 14 covers more information about how you can use this character in DOS commands.

Using the MODE Command to Change Printer Configuration

DOS makes several assumptions about how the printer is connected to the computer and the settings it uses. You can use the MODE command to change the settings the printer uses. The MODE command can affect many aspects of the print process including changing printer configuration, changing the display mode, and changing the serial port configuration. The way you use the MODE

command to change printer settings depends on the type of connection to your printer.

While you may use the MODE command from time to time, you will not use it as frequently as commands like COPY and PRINT. MODE is a command that is frequently used with one hand on the keyboard and the other on documentation such as this book. It is also often included in a batch file so you do not have to worry about entering it correctly each time.

You can display the settings of a device by typing MODE and the device name. It is either LPT1, LPT2, LPT3, COM1, COM2, COM3, COM4, or CON (for the monitor and keyboard). You can also type MODE by itself and DOS will display the mode settings for all connected devices. If you redirect parallel printers (as described later), you will want to include the /Status or /sta switch.

DOS at Work

(Amorette Pedersen is managing editor at Benchmark Productions in Weymouth, Massachusetts. Benchmark specializes in electronic publishing. They use MS-DOS based machines running Ventura Publisher to produce books about using computers.)

The most time-consuming task for us is printing to our Postscript printer. Printing a 500-page book can take several hours of machine time. We wanted to avoid having our systems devoted solely to printing for such long periods. So instead of printing directly from Ventura to the printer, we create printer output files on the hard disk in a directory we've created named OUTPUT. We then exit Ventura and go to the OUTPUT directory. The only files we put in this directory are files we want to print.

Once we're in the Output directory, we use the PRINT command in DOS:

```
PRINT *.* LPT1:
```

and DOS begins printing the files to our printer. Once we've issued the command, we restart Ventura and continue doing page make-up as DOS prints our chapters in the background. It is an incredible time saver.

Changing the Settings of a Parallel Printer Port You can change the configuration of a parallel printer by entering MODE LPT1 (or LPT2 or LPT3). After this, you can enter the number of characters per line, the number of lines per inch and what

happens when a time-out error occurs (a time-out is when the printer does not accept the characters the computer sends it). As an example, you can set the characters per line to 132, the vertical spacing to eight lines per inch and set DOS to not retry sending information when the printer does not accept it, by entering this command:

```
MODE LPT1 COLS=132 LINES=8 RETRY=NONE
```

Like most MODE commands, this can be abbreviated as:

```
MODE LPT1 132,8,NONE
```

The number of columns can be 80 or 132 and the number of lines per inch is 6 or 8. For retry you can use e to return an error, b to return a busy signal, r to return a ready signal, or none to not retry sending information to the printer. The defaults are 80 columns, six lines per inch, and return an error when the printer will not accept information. You do not have to provide all three arguments. You can omit the cols, lines, or retry parameters without preventing DOS from interpreting the other parameters correctly. With the abbreviated version, if you omit one or more arguments, you must include the commas as in MODE LPT1 ,,B to change the retry action.

Changing the Settings of a Serial Printer Port You can change the configuration of a serial port printer by entering MODE COM1 (or COM2, COM3, or COM4). The serial ports are used for communication transmissions such as a modem or a plotter, so the same command applies to setting the serial ports these use. The settings you make for a serial port select how data is transmitted to the printer, plotter, or modem. It is important that DOS send the information in the format that the device expects to receive it. After entering MODE COM1 or another number, you can enter the baud rate, parity, number of data bits, and retry action. As an example, you can set the baud to 2400 bytes per second, the parity to none, the number of data bits to 8, the number of stop bits to 1, and set DOS to not return any signal by entering this command:

```
MODE COM1 BAUD=24 PARITY=N DATA=8 STOP=1 RETRY=NONE
```

This MODE command can be abbreviated as:

```
MODE COM1 24,N,8,1,NONE
```

The number for the baud rate can be one of these numbers:

11	110 baud
15	150 baud
30	300 baud
60	600 baud
12	1200 baud
24	2400 baud
48	4800 baud
96	9600 baud
19	19200 baud

The parity can be n for none, e for even, o for odd, m for mark, or s for space. The number of data bits can be 5, 6, 7, or 8. The stop bits can be 1, 1.5, or 2. The retry action can be the same choices as for parallel ports. The defaults are even parity, seven data bits, 1 stop bit (unless 110 baud is chosen which uses 2 stop bits), and e for retry action. There is no default for the baud rate. These settings must match the serial printer. You will want to check the printer manual for the correct settings to use.

You do not have to provide all five arguments. You can omit the five parameters when you spell them out without preventing DOS from interpreting the other parameters correctly. With the abbreviated version, if you omit one or more arguments, you must include the commas as in MODE COM1 96,,B to change the baud rate to 9600 baud.

Redirecting Output from a Parallel Printer to a Serial Printer Another use of the MODE command is redirecting the output from a parallel printer to a serial printer. This is as simple as entering:

```
MODE LPT1=COM1
```

After you enter this command, output that is directed to LPT1 is redirected to COM1. Before you use this command, you will want to use any MODE COM1 command to configure COM1. Set retry to b for busy. When you are finished, you can return LPT1 to its original destination by entering MODE LPT1 and pressing ENTER.

At a Glance

In this chapter, you have learned about the commands you use to print text. You have learned both how to print the contents of the screen and to print text that is stored in files. You have learned how to complete the following tasks:

- **Print a screen —SHIFT-Print Screen**
- **Print your DOS session—CTRL-Print Screen**
- **Print a file—PRINT or COPY**
- **Add multiple files to the print queue—PRINT filename1 /p filename2**
- **Removes all files from the print queue—PRINT /t**
- **Change printer configuration—MODE**

Reusing Your Command-Line Entries

Entering commands at the DOS prompt is the only way to access the full range of DOS options and features. The disadvantage of command-line entries is that they require more time than Shell selections, which can be made with a quick click of the mouse. To work as productively as possible at the DOS prompt you will want to take advantage of all the help that DOS offers for reusing your command-line entries. DOS stores the previous command entry in a template and allows you to reuse all or part of this template. In addition, you can use DOSKEY to maintain a history of command entries for reuse. You will learn these new techniques in this chapter. Specifically you will learn how to:

What You'll Learn

In this chapter, you will learn how to:

- **Use the function keys to repeat commands**
- **Use DOSKEY to record a complete history of command entries**
- **Enter multiple commands on a command line**
- **Display, change, and reuse the previous commands you used**
- **Record macros with DOSKEY**
- **Execute macros created with DOS key**

Making the Most of Function Keys to Repeat Commands

As you enter a command at the command prompt, your attention is on the fact that DOS is processing your entry but DOS is also performing another task for each entry. DOS automatically stores each command that you enter in a template within memory. Each new command eliminates the previous command from the template replacing it with the new command entry. You can reuse one or more of the characters in this template using the keys shown in Table 10-1.

Table 10-1: Keys for Editing the Command Template

Key	Key's Function
LEFT ARROW or BACKSPACE	Deletes previous character entered at the command prompt
DELETE	Deletes character at cursor's position
INSERT	Inserts characters into the command template without writing over characters in the template
F1 or RIGHT ARROW	Adds the next character in the template to the DOS command prompt
F2	Adds characters from the template from the current character in the template to the character you type after pressing F2 to the DOS command prompt
F3	Adds all remaining characters from the template to the command prompt
F4	Deletes characters in the template from the current template character to the character you type after pressing F4

Executing a Command Again

If you enter a command that looks at the directory of a removable disk such as **DIR A: /P**, you might want to repeat the command for a series of disks. Rather than reenter the command as you replace the disk in the A drive you can use a function key to place the command at the DOS prompt as long as there are no intervening DOS commands. The function key that copies the previous command from the template to the DOS prompt is F3. To execute this command a second time, all you need to do is press ENTER.

F3 is useful if you forget to enter a parameter or switch at the end of a command entry. If you enter DIR A: and the information scrolls off your screen faster than you can read it, press F3 to place DIR A: at the command prompt. Rather than press ENTER immediately you can type a space then **/P** before pressing ENTER. The ease with which you can make the addition removes some of the frustration from your mistake.

Correcting a Spelling Mistake

If you accidentally type the wrong letter in a command, DOS will be unable to process your request and will display an error message. Although you normally can spot the error with a quick glance at the screen, making the correction takes a bit more time if you need to retype the entire entry.

It is not necessary to retype the command. A correction is especially easy when you are replacing an incorrect character since the length of the command-line entry is exactly the same. Using the RIGHT ARROW key to copy characters up to the point of the error or using F2 to copy a group of characters provides the quickest approach to correction. After replacing the erroneous character, F3 can be used to copy the remaining characters from the template to the current line. If you have extra characters you want to remove or missing characters that you want to add, you can use other keys to delete or add characters.

Using the RIGHT ARROW Key or F1 to Copy a Character You can use the RIGHT ARROW key or F1 to copy one character at a time from the template to the command line. With this approach you can copy over as far as the error then make the correction. Suppose you entered the following:

```
DISCCOPY A: B:
```

Since the first C should be a K, you want to copy three characters before making a correction. Follow these steps to make the correction:

1. Press the RIGHT ARROW key three times to copy DIS to the command line.
2. Type **K** to change the spelling of the DOS command.
3. Press F3.

Using F2 to Copy a Group of Characters Besides using F1 for the RIGHT ARROW key to add one character at a time, you can also use the F2 key to add a group of characters from the template to the DOS command prompt. After you press F2, you must then type the character from the template that you want all characters repeated from the current position up to the character that you type. This means that if you type **DIR A:** and you want to check the directory of drive B, you would press F2 and type **A** to repeat all of the characters up to but not including the A. After this you would type **B:** and press ENTER. You can also combine F2 with other keys. Suppose you entered the following:

```
DISKCOPU A: B:
```

Since the U should be a Y, you want to repeat all characters up to the U before making a correction. Follow these steps to make the correction:

1. Press F2 and type **U** to repeat the first seven characters.
2. Type **Y** to change the spelling of the DOS command.
3. Press F3 to repeat the rest of the command.

Adding a Missing Letter

If you forget a character in an entry you can add it to the copy of the command stored in the template. All you need to do is copy the portion of the template entry up to the point of the missing characters to the command line. Next, press INS then type the missing characters before copying the remainder of the template to the command line. As soon as you press INS, DOS stops accessing the template and allows you to place entries on the command line without moving the placeholder it uses to keep track of where it is in the template. When you are finished inserting characters, you can press INS again.

Suppose you enter the following command:

```
COY TEXT B:
```

DOS cannot process the command since it does not recognize COPY with the missing P. To make the correction you would follow this procedure:

1. Press the RIGHT ARROW twice to copy C and O to the command-line.
2. Press INS to allow you to start adding characters without affecting the placeholder in the template.
3. Type **P**.
4. Press F3 to copy the remaining characters from the template. The command-line entry reads **COPY TEXT B:**.
5. Press ENTER to process the corrected command. You do not need to press INS again. When you press ENTER, DOS reverts to replacing the characters in the template with the characters that you type.

Deleting Part of the Original Entry

Typing an extra character causes as many problems as not typing enough **since DOS entries must be exact**. Every time you press the DEL key DOS deletes a character from the template allowing you to reuse the template entry without the erroneous character.

Suppose you accidentally type the following:

```
COPPY TEXT B:
```

To delete the extra P and reprocess the command, follow these steps:

1. Press the RIGHT ARROW key three times to copy COP to the command-line.
2. Press the DEL key to delete the extra P.

3. Press F3 to copy the rest of the template to the command line which now reads as COPY TEXT B:.
4. Press ENTER to reprocess the corrected command.

■ **NOTE:** Pressing the BACKSPACE key deletes a character from the current command-line entry. This action also moves the placeholder one position to the left in the template.

Reusing Part of the Entry

The F2 and F4 function keys can be used to reuse a portion of the entry in the template. The F2 key adds characters from the template to the current entry. The F4 key removes characters from the template so the next character in the template is the one after you deleted. For both keys, you must type the character that you want DOS to repeat on the command line or delete from the template from the template's current position up to, but not including, the character you type. You can use these function keys in combination with the other keys in Table 10-1 to reuse the DOS command stored in the template. To see how this works, suppose you want to perform several DIR commands using different groups of files on drive A and send the output to different files. To try these keys, follow these examples.

1. Type **DIR *.EXE /S /B /O:D >C:\FILE1** and press ENTER to create a file called FILE1 that lists the files in the current directory and any of its subdirectories with a .EXE extension using an abbreviated display and in the order according to their date.
2. Press F2 and type * to repeat the first four characters from the last command.
3. Press F4 and type / to remove *.EXE from the template.
4. Press F2 and type **O** to add /S /B / to the command prompt.
5. Press the BACKSPACE to remove the last /.
6. Press F4 and type **>** to remove /O:D and a space from the template.
7. Press F2 and type **1** to add >C:\ to the command prompt.
8. Type **2** and press ENTER. This creates a file called FILE2 that lists the current directory and its subdirectories in an abbreviated format. The two commands you have entered look like this:

```
DIR *.EXE /S /B /O:D >C:\FILE1
DIR /S /B >C:\FILE2
```

Using DOSKEY to Maintain a History

Using the function keys is convenient when you want to reuse the last command entry. The fact that it only works with the last command is significant since you will often want to access a command after using several intervening commands. The function keys will not work since they cannot access the template for anything but the most recent command processed. The DOSKEY command records the commands you perform and stores them in memory. This lets you access many of the commands you executed previously. You are able to decide how much space you are willing to allocate to previous commands and can determine how much of the command history will be accessible.

In addition to remembering multiple commands, DOSKEY can also create and perform macros. *Macros* are stored keystrokes that will be available for reuse during the current session and which you do not want to keep on a permanent basis the way that you want to keep the commands in a batch file. You can also quickly store commands loaded in the DOSKEY buffer in a batch file which is just like the batch files created in Chapters 12 and 13.

Recording Entries

To have DOSKEY remember your commands, you must first load DOSKEY. To load DOSKEY, type **DOSKEY** and press ENTER. This reserves a buffer of 512 bytes of memory that DOSKEY uses to store your commands. DOS remembers as many previously entered commands as it can store in the buffer. You can use a different buffer by using the switch /bufsize= followed by the number of bytes of the buffer size. If you use several macros you will want to use a larger buffer size to have room to store the macros and additional DOS commands. You can also select the default for insert or overstrike for entering characters at the command prompt by using the /insert or /overstrike switch. The /overstrike switch is the default. The other switch you may want to use is /reinstall, which installs DOSKEY in the computer's memory over the previously installed version of DOSKEY. Once DOSKEY is loaded, it remembers all of your commands; when the buffer becomes full, it shifts the commands so the first command in the buffer is removed.

■ ***NOTE:*** When you use the /reinstall option all macros and keystrokes recorded with the previously installed version of DOSKEY are lost.

Displaying the Command History

DOSKEY has two methods of displaying the commands that are recorded in DOSKEY's buffer. You can press F7 to produce a list of commands in history. Lines will be numbered and the current line will be marked with a >. You can also

type **DOSKEY /HISTORY** and press ENTER to display the list although the entries will not be numbered as shown in Figure 10-1.

You can also reset the list of commands stored in the DOSKEY buffer to remove the commands currently stored. To do this, press ALT-F7. The DOSKEY buffer is also cleared when you restart your system.

Figure 10-1: Commands Listed with DOSKEY

```
C:\>doskey /history
dir C*.*
tree
cd ..
tree
dir c:\agenda\apps\a*.*
cls


C:\>
```

Each command is listed as originally entered.

Changing the Active Command

You can display one of the commands in the DOSKEY buffer at the DOS command prompt to be reused. You can change which of the commands in the DOSKEY buffer is displayed on the command prompt by pressing the UP ARROW or DOWN ARROW keys. The UP ARROW displays the previous command in the buffer according to the list and the DOWN ARROW displays the next command in the buffer. When it gets to the bottom or top of the list, using the UP ARROW or DOWN ARROW starts from the opposite end of the list. You can also display the first or last one in the DOSKEY buffer by pressing PGUP or PGDN. Another method of selecting one of the lines in the buffer to display at the command prompt is to press F9 and type the line number of the command you want. This function key is normally combined with F7 or the /history switch. You can also use the F8 key to repeat a command stored in the buffer. To use F8, type the beginning of the command you want to repeat and press F8. DOS adds the rest of the command in the buffer that has the same beginning as the characters you have typed. This means that if you have a DIR and a PRINT command stored in the DOSKEY buffer, typing **D** and pressing F8 will display the DIR command since it starts with a D. If you want to remove the command that DOSKEY has displayed at the command prompt, press ESC.

Reusing the Active Command

Once you have a command displayed at the command prompt, you can modify it and reuse it. When you alter a command stored in the DOSKEY buffer, you will use keys differently than when you use the DOS template and the first four function keys. Most of the difference between the keys you use is because DOSKEY lets you edit the entire line from any point rather than the DOS template which only treats the prior command as a stream of characters—you can work with backwards and forwards but you cannot jump from one location to another. You can also use these keys to edit the current line when you are making a new entry and you realize you made a mistake at the beginning. This is better than not using DOSKEY since without DOSKEY you would have to use the BACKSPACE or ESC to remove several or all of the characters before you could edit the incorrect ones. The keys you can use to edit commands are listed in Table 10-2. The CTRL-LEFT ARROW and CTRL-RIGHT ARROW use spaces to determine where words are. The INS key, which switches between inserting and overwriting characters on the command prompt, also changes how the cursor appears. When DOSKEY will add characters to the command, the cursor is taller than when DOSKEY will overwrite characters on the command prompt. Pressing ENTER returns DOSKEY to use the default set with the /insert or /overstrike switch. To see how you can edit the lines stored in the DOSKEY buffer using the same commands you used with the template, follow these steps:

1. Type **DIR *.EXE /S /B /O:D >C:\FILE1** and press ENTER.
2. Press the UP ARROW to repeat the command, HOME to move to the beginning and CTRL-RIGHT ARROW to move to *.EXE.
3. Press DELETE six times to remove *.EXE.
4. Press CTRL-RIGHT ARROW twice to move to the beginning of /O:D.
5. Press DELETE five times to remove /O:D.
6. Press END to move to the end, press BACKSPACE to remove the 1, type **2**, and press ENTER to use this altered command.

Table 10-2: Keys for Editing a Command from the DOSKEY Buffer

Key	Key's Function
HOME	Moves cursor to the beginning of the line
END	Moves cursor to the end of the line
LEFT ARROW	Moves cursor one character to the left
RIGHT ARROW	Moves cursor one character to the right

Key	Key's Function
HOME	Moves cursor to the beginning of the line
CTRL-LEFT ARROW	Moves cursor one word to the left
CTRL-RIGHT ARROW	Moves cursor one word to the right
BACKSPACE	Deletes character to the left of the cursor
DELETE	Deletes character at the cursor's position
CTRL-END	Deletes all characters from the cursor's position to end of the line
CTRL-HOME	Deletes all characters from the beginning of the list to the cursor's position
INSERT	Switches between overstrike and insert
ESC	Erases the command

Saving the Command History

You can save the commands that are stored in the DOSKEY buffer. To save these commands, use the redirection character > to send the output of the commands to another file as in DOSKEY /HISTORY >COMMANDS.BAT. This lets you create a file containing DOS commands that you can execute as a batch file. Creating and using batch files are discussed in greater detail in Chapters 12 and 13.

Creating Macros with DOSKEY

DOSKEY lets you create macros. DOSKEY macros have some advantages and disadvantages over batch files (which are described in Chapters 12 and 13). Batch files are also a series of DOS commands—although batch files can contain special DOS commands which provide more features than are available with macros.

Some of the advantages of macros are:

- They are stored in RAM which means they perform faster than batch files although this difference is mostly noticed when you compare macros with batch files stored on floppy disks.
- Macros can run batch files but batch files cannot run macros. However, you can use a batch file to create macros.

Some of the disadvantages of macros are:

- Macros are lost when you reinstall DOSKEY or when you restart your computer while batch files are still available.

- Batch files do not have limits on the number of characters while macros can only contain up to 127 characters.
- You can edit batch files while with macros you must redefine the macro to make any changes.
- You can halt a batch file by pressing CTRL-C once while you must press CTRL-C for every command in a macro to halt the macro.
- Macros cannot contain the special DOS commands such as GOTO, ECHO, IF which let you control the order of commands executed and their appearance on the screen.

Macros work best for performing a group of DOS commands repeatedly during the current DOS session while batch files work best for performing a group of commands during multiple DOS sessions. Also, macros use $0 though $9 for variables where batch files use %0 through %9. Macros also use $L, $G, and $B in place of <, >, and | for redirection. When you lists the macros, you will see the special entries although when they execute the replacement characters will be used. Macros use part of the DOSKEY buffer. The more macros you create the less room the DOSKEY buffer has to store a history of the DOS commands you enter. If you have several macros, you may want to use the /bufsize switch and increase the size of the DOSKEY buffer.

- ***NOTE:*** If you use macros frequently, use a batch file to define your favorite macros so you can run the batch file and have all of the available.

Recording a Macro

To create a macro, enter DOSKEY, the macro name, an equal sign, and the DOS commands you want the macro to perform. Since macros can perform several commands, you need to separate the different commands by entering $t. An example of creating a macro is:

```
DOSKEY BU=DIR /O:D >FILEDATE $T PRINT FILEDATE $T XCOPY *.* A: /P
```

This creates a macro called BU that lists the files in the current directory according to their date and stores the output in a file called FILEDATE. Next this macro prints the FILEDATE file and then the XCOPY command prompts for each file if you want to copy the file.

Once you create a macro, you have two options for changing it. You can redefine the macro and let the new macro definition replace the old one. Another

option is, if the DOSKEY command that created the macro is still in the DOSKEY buffer you can display the line that defined the macro and edit it.

You can also display the defined macros by entering DOSKEY /MACROS. This displays a list like the one in Figure 10-2.

Figure 10-2: Displaying a Macro

```
C:\>DOSKEY /DMACS
LOOK=DIR A: /P
DISP=TYPE C:\MENU.BAT
PRT=PRINT c:\MENU.BAT
PRT2=DIR C: $g PRN $t TREE $g PRN $t DIR A:

C:\>
```

&g represents > and $t is the separator betweens commands.

To delete a macro, enter DOSKEY, the macro name and an equal sign with nothing after it, as in:

```
DOSKEY BU=
```

This removes the macro from memory. You can also delete all macros by pressing ALT-F10.

Running a Macro

After the macro is created, you are ready to run it. To run a macro, type the name and press ENTER. As described in the note, you do not want anything between the DOS command prompt and the macro name. If the macro uses parameters you must add a space and the parameter after the macro name. You will see the **Bad command or filename** message if DOS cannot interpret your entry as a DOS command or executable file name and DOSKEY cannot interpret your entry as a macro name.

- ***NOTE:*** If you usually use a DOS command with specific switches or parameters, you can create a macro with the same name as the macro command. An example is DOSKEY DIR=DIR /O:N to consistently list the current directory in alphabetical order. If you later want to run the DOS command without running the macro, type a space before typing the command since DOSKEY will not recognize it as a macro name because the first character is a space. You can also use this feature

with other software by having the macro change the drive and directory and then load the program.

Saving the Macro

Saving macros lets you save them to a file so you can edit them and use them later in a batch file. As described above, you can save a macro or the commands in the DOSKEY buffer by using the /history switch and redirecting the output to a file. If you only want to save the macros in a file, use the /macros switch as shown here:

```
DOSKEY /MACROS >MACROS.BAT
```

This stores all of the macros in a file called MACROS.BAT. Any special DOSKEY characters such as $t are translated into their DOS equivalents. To convert this file into a useable batch file, you will need to put command separated ¶ characters onto separate lines.

Using Parameters in the Macro

The DOSKEY macros can use parameters which means that when you run the macro, after the macro name you can enter specific information that the macro will use in the macro. This information can include the files the DOS commands in the macro will use. These parameters are labeled $0 through $9. When you define the macro, you will type these parameters in the macro, as in COPY $1 A:. The parameters $1 through $9 are assigned the first through ninth words (characters separated by spaces) entered after the macro name. The $* is a special parameter which represents all of the text after the macro name.

The $* parameter assigns all of the text after the macro name to the parameter. You can use this to provide multiple DOS command parameters and switches. An example of a macro definition using this parameter is:

```
DOSKEY BU=BACKUP $* /M
```

This macro definition creates a BU macro that uses whatever parameters are provided when the BU macro is run in for the BACKUP command in addition to the /M switch. When you run the macro, some of the ways you might enter this command are as follows:

```
BU C: A: /S
BU *.EXE A:
```

If you do not use the $* parameter in a macro definition, you can use nine separate parameters. This lets you supply different parameters for one or more commands stored in the macro. When you run the macro, DOS assigns the first word after the macro name to the $1 parameter, the second to $2 and so on until all parameters are assigned or all words after the macro name are assigned. An example of a macro definition using several parameters is:

```
DOSKEY BU=RENAME $1 $2 $T COPY $2 A: $T ATTRIB A: $2 /R+
```

This macro uses the first parameter for the original filename and the second parameter for the altered filename. This macro renames the file, copies it to drive A, and changes the file so it is read-only. When you run the macro, you could make these entries:

```
BU MY_FILE MY_FILE.DOC
BU *.DOC *.DAT
```

Redirecting Macro Input and Output

As mentioned before, macros use different symbols for redirecting input and output. This is because you cannot include these symbols in a DOSKEY command without redirecting input or output for that DOSKEY command. A macro uses $L in place of a less-than sign to accept input from another source. A macro can also use $G in place of a greater-than sign to send output to another source. When you want to append data to a file with two greater than signs, you would enter gg. Finally, the vertical bar character to pipe input from one command to another is represented by $b. If you want to use a dollar sign in other context in a macro, you must use two dollar signs as in $$. Some examples of macro definitions using these characters are:

```
DOSKEY DA=DIR A: $g MY_FILES
DOSKEY SORT =$L %1
DOSKEY DPAGE=MORE $b DIR
DOSKEY DLIST=DIR A: $G DISKS $T DIR B: $G$G DISKS
```

At a Glance

In this chapter you learned all the techniques for reusing DOS command entries. Mastering these techniques can make DOS command-line entries almost as quick as selecting commands with the DOS Shell. The new productivity techniques you discovered include:

- **Enter the previous command at the DOS prompt—F3**
- **Copy characters from the previous command up to a specific character—F2**
- **Retype the previous command a character at a time—RIGHT ARROW or F1**
- **Skip characters in the previous entry—F4**
- **Invoke the DOSKEY program —DOSKEY**
- **Display commands recorded by DOSKEY—F7**
- **Display matching command from history—F8**
- **Display a specific command number—F9**
- **Create a macro—DOSKEY macro name=macro commands**
- **Running a DOSKEY macro—macro name**
- **Delete a macro—DOSKEY macro name =**
- **Delete all DOSKEY macros—ALT-F10**
- **Add replaceable parameters to macros—$0 through $9**

Working with the DOS Editor

The Editor provides a simple text editor that you can use to create text files. Text files only contain the data that you enter without the addition of special codes. Text files can be used for many purposes including creating a simple memo, creating test data for an application, storing DOS batch file instructions, and writing programs. While you can use a word processor to create and edit text files, you do not need to purchase any additional software to create them since you can just as easily use the System Editor that comes with DOS 5. For writing letters, reports, and other lengthy documents, a word processing package offers enough additional advantages to be worth the purchase price. The System Editor does not provide the word processing features such as spell checking and fancy print capabilities.

What You'll Learn

After reading this chapter you will be able to handle the following tasks:

- **Use EDLIN's basic features**
- **Start the System Editor from the Dos Shell or DOS prompt**
- **Select menu options for the Editor and complete dialog boxes**
- **Enter and edit a text file with the editor**
- **Save the text file**
- **Print the text file**
- **Make customizing changes to the editor screen**

Advantages Over EDLIN

The DOS Editor is DOS 5's replacement for EDLIN, its predecessor. The Editor offers many advantages over EDLIN and you will want to make the transition even if you are an experienced EDLIN user.

EDLIN is the line editor included with DOS 5 as well as earlier releases of DOS. While it was useful to have an editor which you could use to create simple files, EDLIN does not provide features beyond the simplest entry and edit capabilities. EDLIN does not have a menu so you must know the command you want and how to invoke it. Without a help system, you must have your book open as you try to master the program because the commands are single letters with parameters for specifying line numbers that should be affected and other important information.

Also, EDLIN is a line editor—which means you work with a file *a line at a time*. For example, if you wanted to edit the fifth line, you would type a 5 to edit that line rather than the simpler method used by the System Editor and word processors that allows you to move to any line on the screen and make a change. The Editor, on the other hand, is a full screen editor. You can forget about line numbers because the Editor will let you move from one line to another simply by pressing an arrow key. Since the menu is on top of the screen, you do not have to have its feature set memorized before you can use it. The menu will prompt you with the information it needs so you do not have to worry that DOS will misinterpret your request.

■ ***NOTE:*** If you are running EDLIN or the EDITOR without a hard disk you will need to insert the correct disk before attempting to run it. For the Editor, insert the disk labeled BASIC/EDIT. For EDLIN, insert the disk labeled STARTUP/SUPPORT (5.25) or Support (3.50).

To start EDLIN, type **EDLIN** at the DOS prompt. Unless you have the DOS directory as part of your path statement you will want to activate the DOS directory first. Table 11-1 shows the commands EDLIN can use with an example. You will want to use the table as a reference if you ever need to use EDLIN or if you already use EDLIN but are not familiar with all of the commands. You can also type a ? and press ENTER when EDLIN displays a new file with an asterisk on the next line. EDLIN will display a list of all the commands it supports. When you are finished with EDLIN, type an **e** to exit and save or a **q** to quit.

Working with the Editor

The Editor takes a little longer to load into memory than EDLIN but once it is loaded it will save a significant amount of time as you work with text files. Menu commands are easier to remember than the single letter approach used with EDLIN and the full screen entry allows you to make changes in several places quickly.

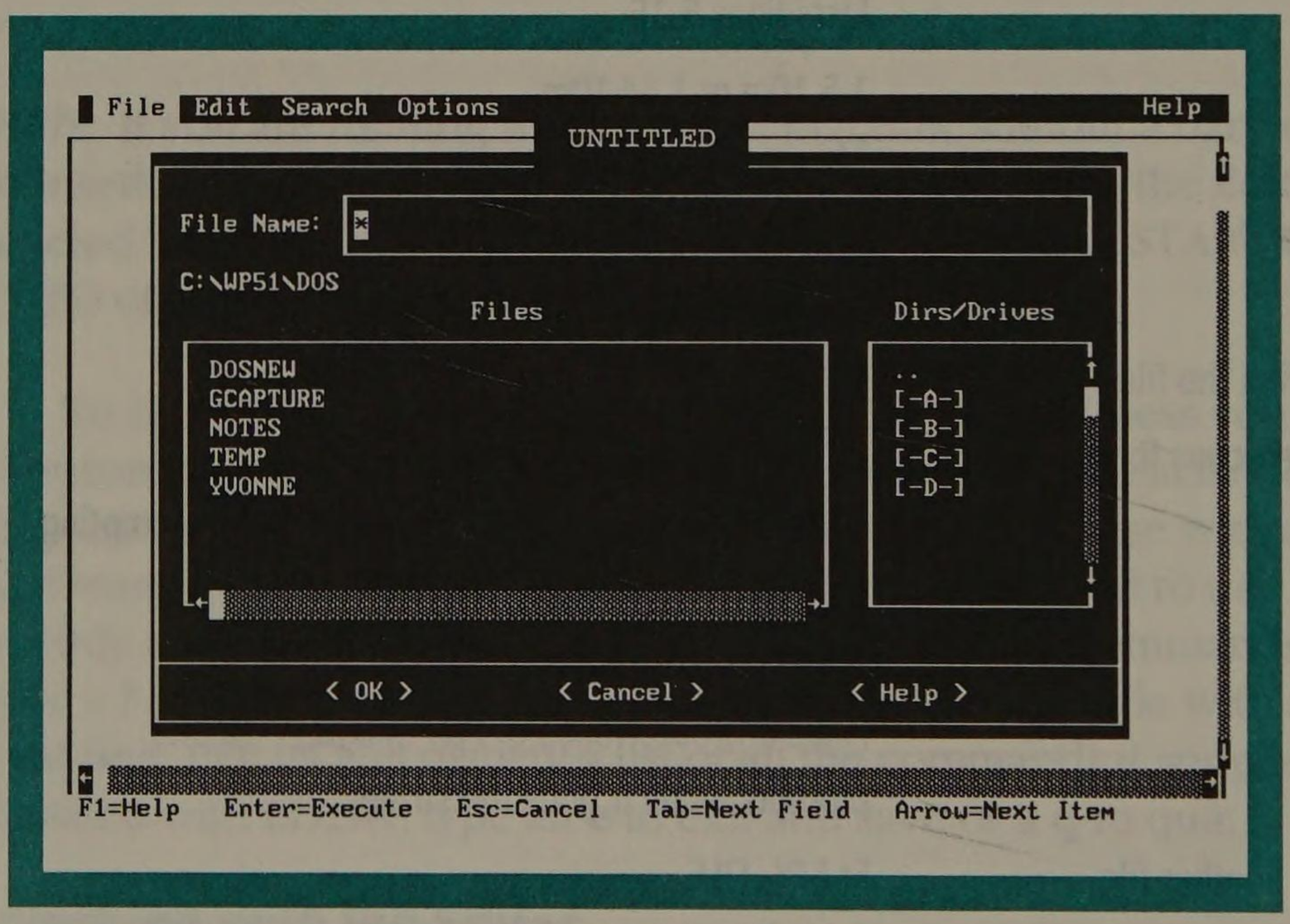
Table 11-1: EDLIN commands

Key	Command Function	Example
#	Displays a line to edit	3
a	Loads a portion of a large file	50a Loads the next 50 lines
c	Copies lines to a new location	1,3,6,3c Copies lines 1-3 three times starting at line 6
d	Deletes lines	1, 4d Deletes lines 1-4
e	Exits after saving the file	e
i	Inserts lines until you press CTRL-C	5i Starts inserting lines at line 5
l	Lists a block of lines	5,15l Lists lines 5-15
m	Moves lines	1,5,10m or 1,+4,10m Moves lines 1-5 to line 10
p	Pages through the file	50,20p Displays 20 lines on a page starting at line 50
q	Quits without saving the file	q
r	Replaces text it searches for	1,10 ?r PR1 ^Z PR2 Replaces PR1 with PR2 in lines 1-10 after prompting for confirmation
s	Searches for text	1,10 ?s PR1 Finds any occurrence of PR1 in lines 1-10 and prompts when it finds the string
t	Transfers lines to another file	5t MY_FILE Adds the contents of MY_FILE starting at line 5
w	Writes a block of lines to the disk	50w Writes the first 50 lines to the file

Starting the Editor

Starting the Editor is easy since you can do it from the DOS command prompt or from the DOS Shell. To start the Editor from the DOS command prompt, type **EDIT** and the name of the file you want to use. You can also enter **EDIT** without a filename to load the Editor without loading a file. You can start the Editor from the DOS Shell by highlighting MS-DOS Editor and pressing ENTER or double clicking it on the screen. You will be prompted for the name of the file you want to edit. You can either enter a filename and press ENTER or just press ENTER to start the Editor without loading a text file. If the Editor is loaded without a file, the screen will look like Figure 11-1. You can press ENTER from the initial Editor screen to see the Survival Guide or press ESC to clear this dialog box. Later you will learn how you can use the help window to display different information about using the Editor. When you press ESC, the dialog box will disappear from the screen and you can create your own text files.

Figure 11-1: Editor Screen Without a File Selected



You can type a filename or select one from the list.

Screen Elements The Editor has several screen elements that you should be familiar with after working with the DOS Shell. The menu bar is at the top of the screen with the selectable choices displayed. Below the menu bar is the name of the file you are using or the word Untitled if the file is not named yet. The reference bar is at the bottom of the screen. It contains information such as the name of the

application (MS-DOS Editor), the column and row of the cursor, any buttons you may want to select with the mouse such as <F1=Help>, and any directions for your current actions. The rest of the screen area is for editing your text. The bottom and right edge of the screen contain the scroll bars which you can use to judge how far you are from the top or bottom of the file. You can also use the scroll bar to move to another location.

Options Used When Starting the Editor The Editor has a few switches you can use when you start the program. If you use the /b switch, the Editor will use monochrome colors (black and white) rather than colors. If you are using a CGA monitor, you may want to use the /G switch to have DOS refresh the screen more frequently and diminish the "snow" or flicks of white on the screen. If you are using an EGA or VGA monitor, you may want to use the /h switch so the Editor uses the maximum number of lines the hardware allows. On EGA screens this is 43 lines and on VGA it is 50 lines. Finally, you may want to use the /nohi switch if your monitor cannot display the high intensity colors. Normally, the Editor screen will display the bottom text and the first letters of menu selections when the menu is active using an intense white. When you use the /nohi switch, the Editor uses black for the text on the bottom and red for the first letters of the menu selections when the menu is active. You can also use these switches if you start the Editor from the DOS Shell by using the File Properties command and supplying the switches you want in the Command text box. You can review the specific File Properties features in Chapter 5.

Using the Editor's Menus and Dialog Boxes

Once the Editor screen is displayed, you can begin typing the text you want to store. The menu commands provide features to help you create a file quickly. Most of the skills you used to make menu selections with the DOS Shell program also apply to the Editor.

Activating the Menu and Making a Selection If you want to use the keyboard for selections you can activate one of the items in the menu bar by pressing the ALT key and then typing the highlighted letter. With a mouse, you can just point to it and click on the menu name you want to activate. The mouse in the Editor displays as a small rectangle called the *mouse cursor*. In Figure 11-1 the mouse cursor is on the E in Enter. Once a menu is displayed, you can select one of the options by either typing the letter or clicking the menu option. You can also select menus and menu options by using the arrow key to highlight the one you want and pressing ENTER. When you want to back out of menu selections you have made, press ESC.

Supplying Information Through the Dialog Boxes Dialog boxes often appear when you are making a menu selection. If you have been working with the DOS Shell you will already be familiar with the methods used to enter data and make selections with these boxes. Like DOS Shell dialog boxes, the dialog boxes in the Editor may have text boxes to enter text, option buttons to select one of several options, check boxes to add or select features, or lists to select files. In text boxes the Editor expects you to enter text such as a filename or the text you want the Editor to find.

To use option buttons, you must select an option by using the arrow keys to move the bullet that marks the current selection. In check boxes you must press the space bar after moving the highlight to the check box text before the feature you want to add or remove. In lists, you must highlight the option you want. You can select the part of the dialog box you want by holding down the ALT key and typing the highlighted letter. You can also switch from one item in the dialog box to another by pressing TAB or to the previous item by pressing SHIFT-TAB. With a mouse, you just have to point to the item you want and click the mouse. In a list, you can use scroll bars to change the part of the list that appears. You can click above or below the scroll box that indicates your current position to shift the list in the chosen direction. You can also drag the scroll box to a new location to shift the part of the list to the location indicated by the scroll box's position on the scroll bar.

Most of the dialog boxes have three buttons labeled OK, Cancel, and Help. Pressing ENTER selects the OK button. Pressing ESC selects the Cancel button. Pressing F1 selects Help. Of course, you can click on any of these boxes with your mouse to make a selection.

Using Function Key Combinations to Invoke Features At the bottom of the Editor screen, you will see several key selections you can make. You can either select the options these present by pressing the key combinations or by pointing to them with the mouse and clicking the mouse.

Getting Help in the Editor

The Editor provides built-in help which makes it easier to use than EDLIN. You do not have to turn to the manual as you learn to use its features since a brief summary of its features are provided on help screens.

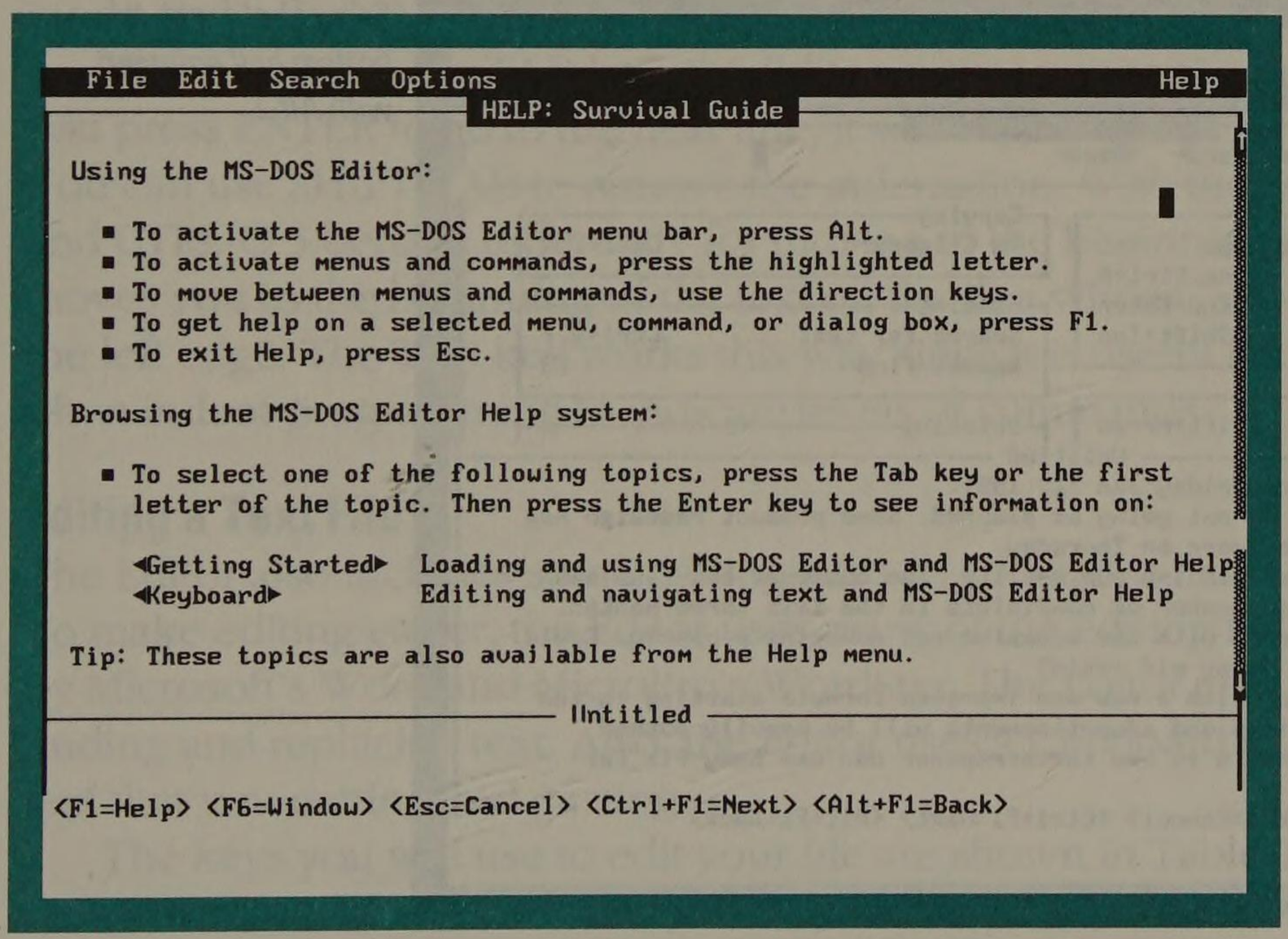
Activating The Help Screen You can quickly access the help information when you first enter the Editor by pressing ENTER at the original dialog box. At other times you can display help information by pressing F1 to open a help window like the one in Figure 11-2. Another method is pressing ALT-H to display the Help menu. A third option is pressing SHIFT-F1 which always displays the Getting Started help

topic. The help window remains open until you press ESC or click <ESC=Cancel> in the reference bar.

Making Use of the Context-Sensitive Help Features The help information in the Editor is *context sensitive*, which means the Editor displays the help information for the topic it thinks you need based on your current actions. While you are using a help window, you can press ALT-F1 or select the Back help topic to display a help topic you previously viewed (the Editor remembers the last 20 topics). Some of the help topics are in order so you can press CTRL-F1 to display the next help topic or SHIFT-CTRL-F1 to display the previous help topic. Another method of selecting a topic to display more information is by activating one of the displayed help topics which are enclosed in brackets. You can point to a topic and click on the topic or you can move the cursor to the topic you want and press ENTER.

To move between help topics on the help window, press TAB or SHIFT-TAB to move forward or back through the available help topics. You can also type a character or press SHIFT and a letter to move to the next or previous help topic that starts with the letter. The last option is using the arrow keys to move the cursor. While this last method works, it is not as efficient as the other methods.

Figure 11-2: Help Window


File Edit Search Options Help
HELP: Survival Guide
Using the MS-DOS Editor:
■ To activate the MS-DOS Editor menu bar, press Alt.
■ To activate menus and commands, press the highlighted letter.
■ To move between menus and commands, use the direction keys.
■ To get help on a selected menu, command, or dialog box, press F1.
■ To exit Help, press Esc.
Browsing the MS-DOS Editor Help system:
■ To select one of the following topics, press the Tab key or the first letter of the topic. Then press the Enter key to see information on:
◄Getting Started► Loading and using MS-DOS Editor and MS-DOS Editor Help
◄Keyboard► Editing and navigating text and MS-DOS Editor Help
Tip: These topics are also available from the Help menu.
Untitled
<F1=Help> <F6=Window> <Esc=Cancel> <Ctrl+F1=Next> <Alt+F1=Back>

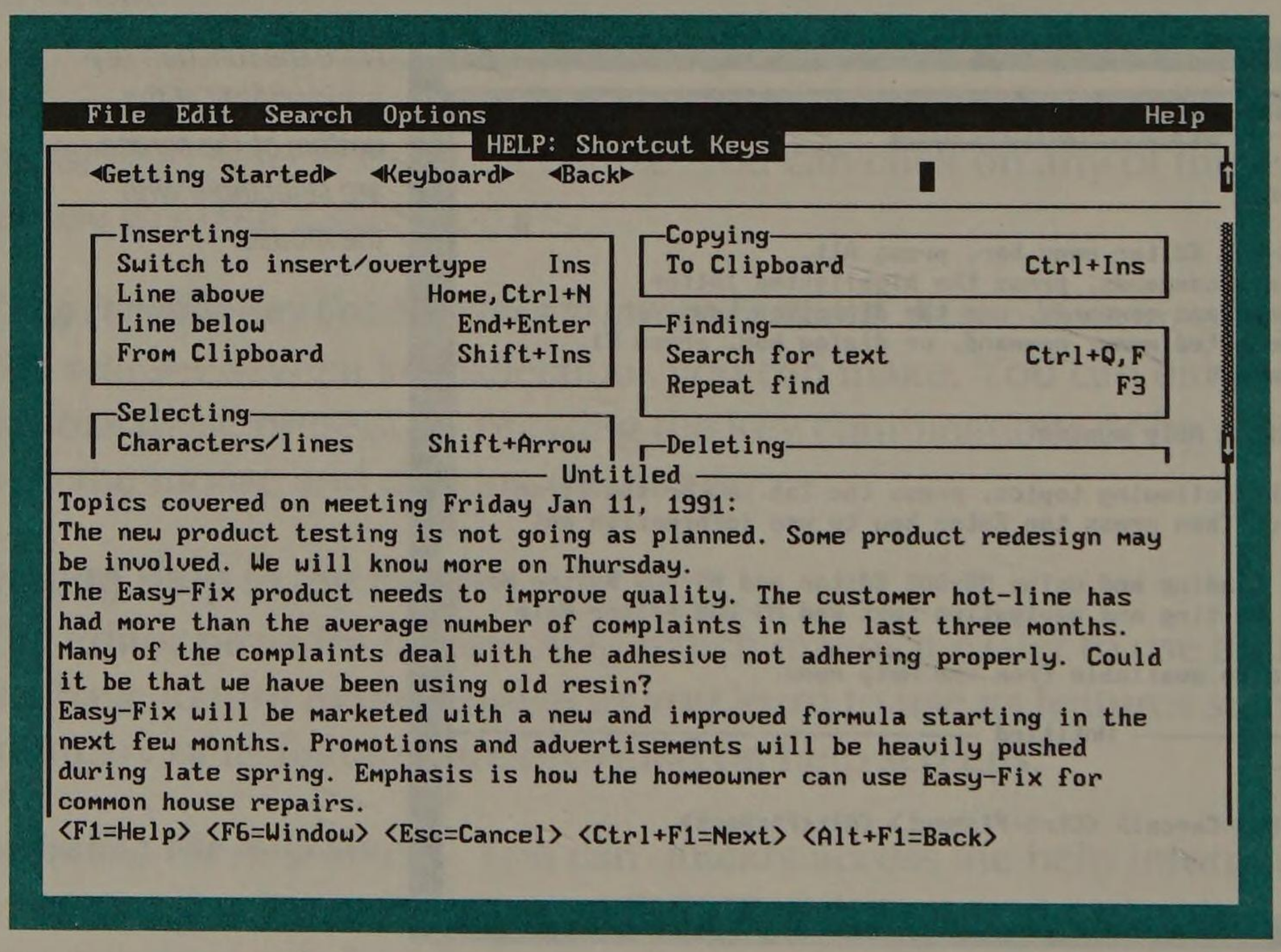
Note the function key assignments at the bottom of the screen are selectable with the mouse.

■ ***NOTE:*** Many of the Editor commands for your text files also work with the Help window. For example, you can use the Editor commands to search the help

information for a particular word or you can use the Editor commands to print a help topic.

Changing the Help Window The Editor can have two types of windows. It can have a Help window to display help information and it can have a window for modifying your file. When you press F1, you open a help window that initially occupies most of the screen. You can switch between a help window and a file window by pressing F6. This lets you maintain help information on the screen while you edit your file. You can also change the proportion of the screen the help window and the file window uses. When you press ALT-+, you increase the current window by one line. When you press ALT--, you decrease the current window by one line. Figure 11-3 shows two windows displayed. When a help topic contains more information than it can fit onto a screen, you may want to scroll through the information by using the UP ARROW and DOWN ARROW. The same keys to move the cursor (and which you will learn to use for editing the text in the file window) can be used to change the section of the help topic that appears in the Help window.

Figure 11-3: Both Help Information and Edit Window on the Screen


File Edit Search Options Help
HELP: Shortcut Keys
◄Getting Started► ◄Keyboard► ◄Back►
Inserting
Switch to insert/overtype Ins
Line above Home,Ctrl+N
Line below End+Enter
From Clipboard Shift+Ins
Copying
To Clipboard Ctrl+Ins
Finding
Search for text Ctrl+Q,F
Repeat find F3
Selecting
Characters/lines Shift+Arrow
Deleting
Untitled
Topics covered on meeting Friday Jan 11, 1991:
The new product testing is not going as planned. Some product redesign may be involved. We will know more on Thursday.
The Easy-Fix product needs to improve quality. The customer hot-line has had more than the average number of complaints in the last three months. Many of the complaints deal with the adhesive not adhering properly. Could it be that we have been using old resin?
Easy-Fix will be marketed with a new and improved formula starting in the next few months. Promotions and advertisements will be heavily pushed during late spring. Emphasis is how the homeowner can use Easy-Fix for common house repairs.
<F1=Help> <F6=Window> <Esc=Cancel> <Ctrl+F1=Next> <Alt+F1=Back>

Screen split with the bottom half enlarged with Alt-+.

■ ***NOTE:*** If you use the /h switch to start the Editor and you are unfamiliar with the keys you may want to use, keep the help screen displayed and use ALT-+ and

ALT-- to change the window size so you can see the help information you want as you edit your file.

Creating a Text File with the Editor

Creating a text file is as simple as typing the characters you want. As you type, the Editor will indicate the column and row position of the cursor in the lower right corner. Each line can contain up to 256 characters. As you type more characters than can fit across the screen, the Editor scrolls your text file to the left or right in order to display the portion of the file you are editing. Since the Editor does not move characters to the next line the way a word processor does, you will need to press ENTER when you want to start the next line. As you use more lines than can fit on the screen, the Editor scrolls your text file to display the portion of the file you are editing.

When you type characters, they are added at the cursor's position and the cursor moves to the next space to the right. Edit has Insert and Overwrite modes which control what happens when you type new characters in the middle of a file. During Insert mode, the default, the characters you type are added and the characters to the right are shifted further to the right. During Overwrite mode, the characters you type replace the characters to the right. You can switch between the two modes by pressing INSERT or CTRL-V. The cursor is a small solid line in insert mode and a flashing rectangle in Overwrite mode.

When you use the TAB key, the Editor moves you to the next tab stop. When you press ENTER to go to the next line, it will be indented like the line above it. You can use SHIFT-TAB to remove the indentation. With the indentation, HOME and CTRL-Q S behave differently for moving to the beginning of the line. HOME moves you to the beginning of the text on the line while CTRL-Q S moves you to the left edge. The TAB key works this way since it is useful for programmers who often indent programming to indicate levels of commands.

Editing a Text File

The Editor also includes some editing keys so you can quickly alter your text file. To make editing easier, the Editor uses many of the editing key combinations used by Microsoft's Word and MicroPro's WordStar. The Editor also includes options for finding and replacing text. Also, the Editor uses a Clipboard to temporarily store text that you cutting and pasting.

The keys you will use to edit your file are shown in Table 11-2. These keys move the cursor to another location. Once you position the cursor, you can enter new characters. With a mouse, you can point to where you want to move the cursor and click the mouse. You may want to use the scroll bars to change the por-

tion of the file that appears in the window. To change the part of the file on the screen, point to a position on the scroll bar and click it. This moves the file in the direction you indicate.

While editing you will also use the arrow and other keys to select text. Basically, move to the beginning of the text you want to select and hold down the SHIFT key as you press the arrow keys to move to the end of the text you want to select. You can also select text with a mouse by clicking the beginning of the text to select and then dragging the mouse to the opposite corner of the text to select. *Dragging* the mouse means holding down the mouse button as you move it across the screen. When you select text from a help window, you select a line at a time rather than characters from a line.

Table 11-2: Keys for Editing Text

Action	Key to Press	WordStar Equivalent
Move cursor up one line	UP ARROW	CTRL-E
Move cursor down one line	DOWN ARROW	CTRL-X
Move cursor one character to the right	RIGHT ARROW	CTRL-D
Move cursor one character to the left	LEFT ARROW	CTRL-S
Display the next screen of text	PGDN	CTRL-C
Display the previous screen of text	PGUP	CTRL-R
Move the cursor one word right	CTRL-RIGHT ARROW	CTRL-F
Move the cursor one word left	CTRL-LEFT ARROW	CTRL-A
Move to the beginning of the line	HOME	CTRL-Q S
Move to the end of the line	END	CTRL-Q D
Move to the beginning of the next line	CTRL-ENTER	CTRL-J
Move to the beginning of the file	CTRL-HOME	CTRL-Q R
Move to the end of the file	CTRL-END	CTRL-Q C
Move to top of the window		CTRL-Q E
Move to bottom of the window		CTRL-Q X
Scroll up one line	CTRL-UP ARROW	CTRL-W
Scroll down one line	CTRL-DOWN ARROW	CTRL-Z
Scroll left one screen	CTRL-PGUP	

Action	Key to Press	WordStar Equivalent
Scroll right one screen	CTRL-PGDN	
Insert a new line at cursor's position		CTRL-N
Insert line and move to the beginning of it	ENTER	CTRL-M
Delete character to the left	BACKSPACE	CTRL-H
Set markers (up to)		CTRL-K then 0-3
Go to marker		CTRL-Q then 0-3
Delete selected characters or character to the right	DELETE	CTRL-G
Delete to end of word	CTRL-T	
Delete to end of line		CTRL-Q Y
Delete entire line		CTRL-Y
Delete leading spaces from selected areas	SHIFT-TAB	
Switch between Insert and Overwrite modes	INSERT	CTRL-V

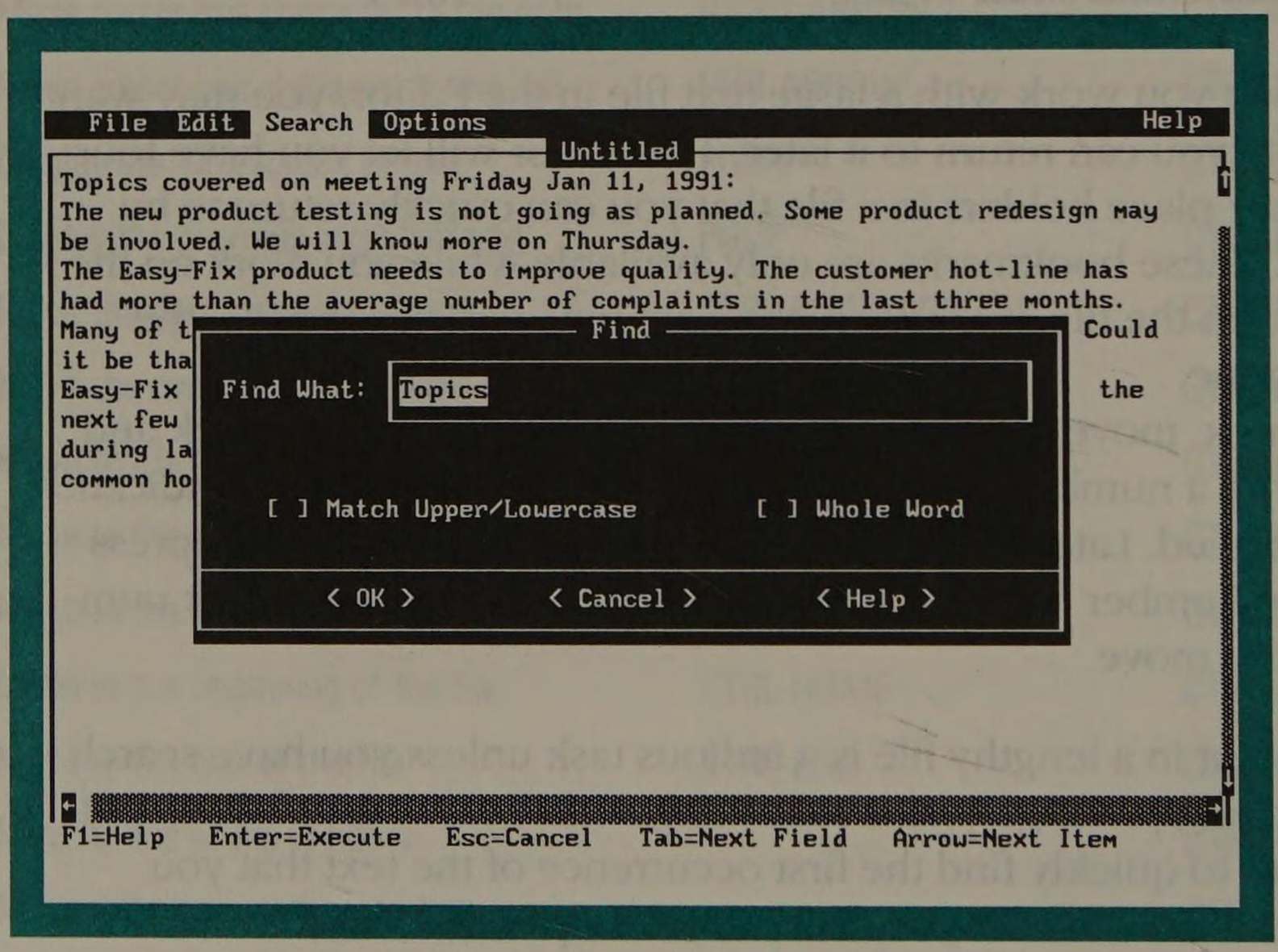
Using Bookmarks When you work with a large text file in the Editor, you may want to mark your place so you can return to it later. The Editor will let you have four bookmarks, which are place holders in a file that you can quickly return to by pressing a few keys. These bookmarks are only available while you work on the file. When you work on the file and save it, the next time you use the file, the bookmarks will be gone.

To add a bookmark, move the cursor to where you want the bookmark and press CTRL-K and type a number between 0 and 3. You will not see any indication that a bookmark is placed. Later when you want to return to a bookmark, press CTRL-Q and type the number between 0 and 3 representing the bookmark number where you want to move.

Finding Text Finding text in a lengthy file is a tedious task unless you have search capabilities in the editing package you are working with. Search capabilities put the computer to work to quickly find the first occurrence of the text that you specify. The Editor can find text for you and provides options to select whether the match must be the same case and whether the text you are looking for must be an entire word. While the example uses the text in the file you are editing, you can also use this to find words in the Help window by using the Search Find command from a Help window.

To search for text, move the cursor to the beginning of where you want the Editor to begin its search. Next, select Search from the menu bar (by pressing ALT-S or clicking Search) and selecting Find. You can also select this by pressing the WordStar equivalent of CTRL-Q and then an F. The Find dialog box in Figure 11-4 appears. Notice that the Editor fills the Find What text box with the word at the cursor's location. This is the case when you have not selected text before using this command. You can type a new entry (which will replace the current one) or use the word the Editor displays. If you want the text to match upper- and lowercase, you will want to press TAB and the space bar or click the Match Upper/Lowercase check box. If you want the text you are searching for to be the entire word, press TAB to move to the Whole Word check box and press the space bar or click the Whole Word check box. As an example, when this check box is selected and you are looking for word "the," the Editor will only find the word the. If the check box is not selected, the Editor will find words such as other, these, they, and rather since they contain the three-letter combination "the."

Figure 11-4: Find Dialog Box

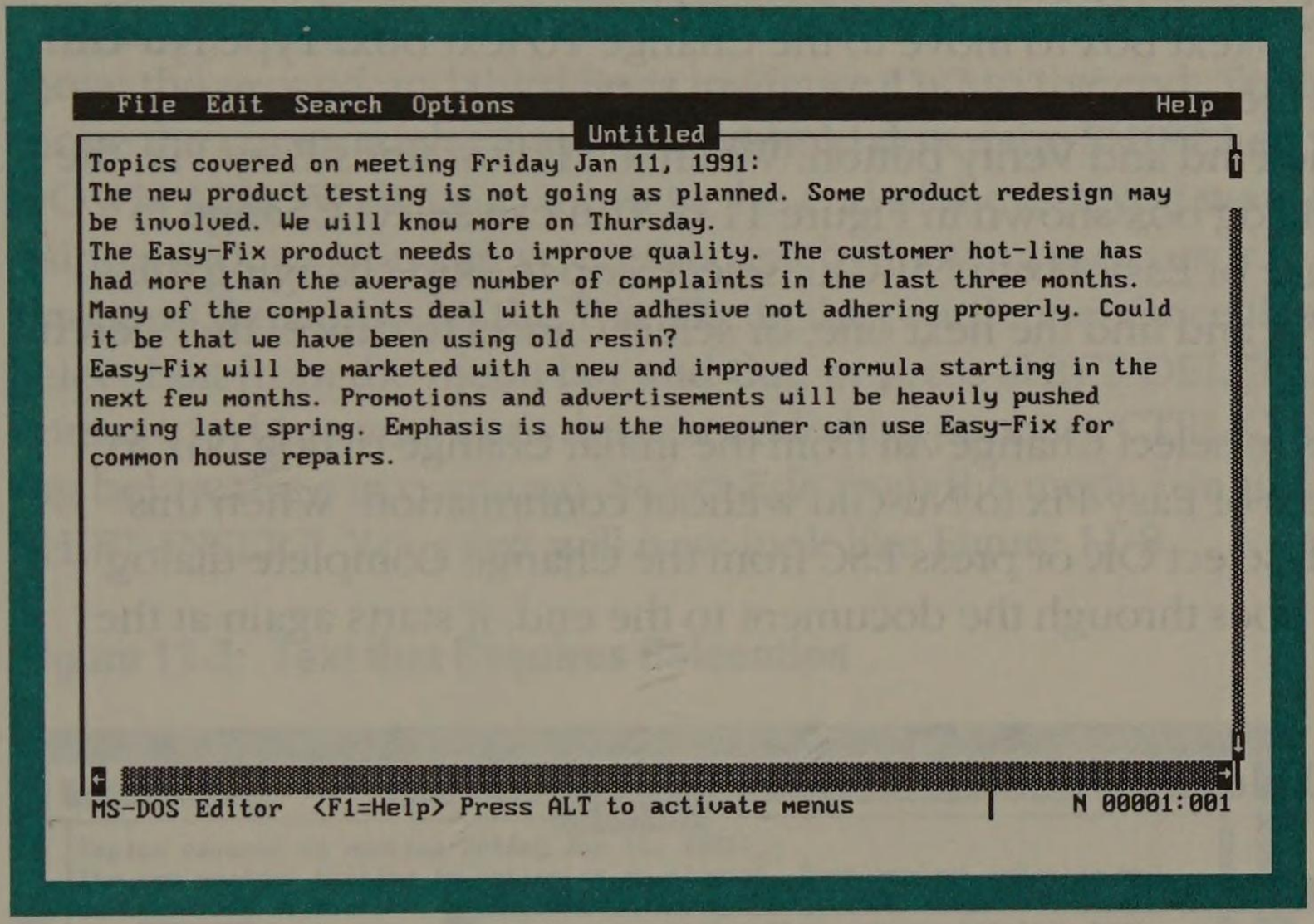

File Edit Search Options Help
Untitled
Topics covered on meeting Friday Jan 11, 1991:
The new product testing is not going as planned. Some product redesign may be involved. We will know more on Thursday.
The Easy-Fix product needs to improve quality. The customer hot-line has had more than the average number of complaints in the last three months.
Many of t
it be tha
Easy-Fix
next few
during la
common ho
Could
the
Find
Find What: Topics
[ ] Match Upper/Lowercase
[ ] Whole Word
< OK >
< Cancel >
< Help >
F1=Help Enter=Execute Esc=Cancel Tab=Next Field Arrow=Next Item

Your search entry in the Find What? box can be as long as 127 characters.

As an example, suppose you want to check the text shown in Figure 11-5 to find any occurrence of the word quality. First you would select Search and Find or press the WordStar equivalent of CTRL-Q and an F. Next you would type **quality**

and press ENTER or click the OK button. When the Editor finds a match, the Editor selects the text you are searching for. If the Editor does not find a match, the Editor would display the Match Not Found dialog box. When you want to find the next occurrence of the text, you can select Search and Repeat Last Find or press F3 or CTRL-L (the WordStar equivalent). The Editor will move to the next occurrence of the text. If no more occurrences exist, the cursor will not move. After the Editor goes through the document to the end, it starts again at the beginning.

Figure 11-5: Text to Check for the Word "quality"


File Edit Search Options Help
Untitled
Topics covered on meeting Friday Jan 11, 1991:
The new product testing is not going as planned. Some product redesign may be involved. We will know more on Thursday.
The Easy-Fix product needs to improve quality. The customer hot-line has had more than the average number of complaints in the last three months. Many of the complaints deal with the adhesive not adhering properly. Could it be that we have been using old resin?
Easy-Fix will be marketed with a new and improved formula starting in the next few months. Promotions and advertisements will be heavily pushed during late spring. Emphasis is how the homeowner can use Easy-Fix for common house repairs.
MS-DOS Editor <F1=Help> Press ALT to activate menus
N 00001:001

You can change all occurences or selectively decide.

Replacing Text Another convenient feature of the Editor is the ability to find text and replace it with different text. This feature is useful if you want to send the same note to two people with just a few changes or if you want to change a drive designation or filename in a text file that contains batch commands.

The replace feature has options to select whether the match must be the same case. Just as with search you can also specify whether the text you are looking for must be an entire word. You also can confirm each replacement or have the replacements performed automatically.

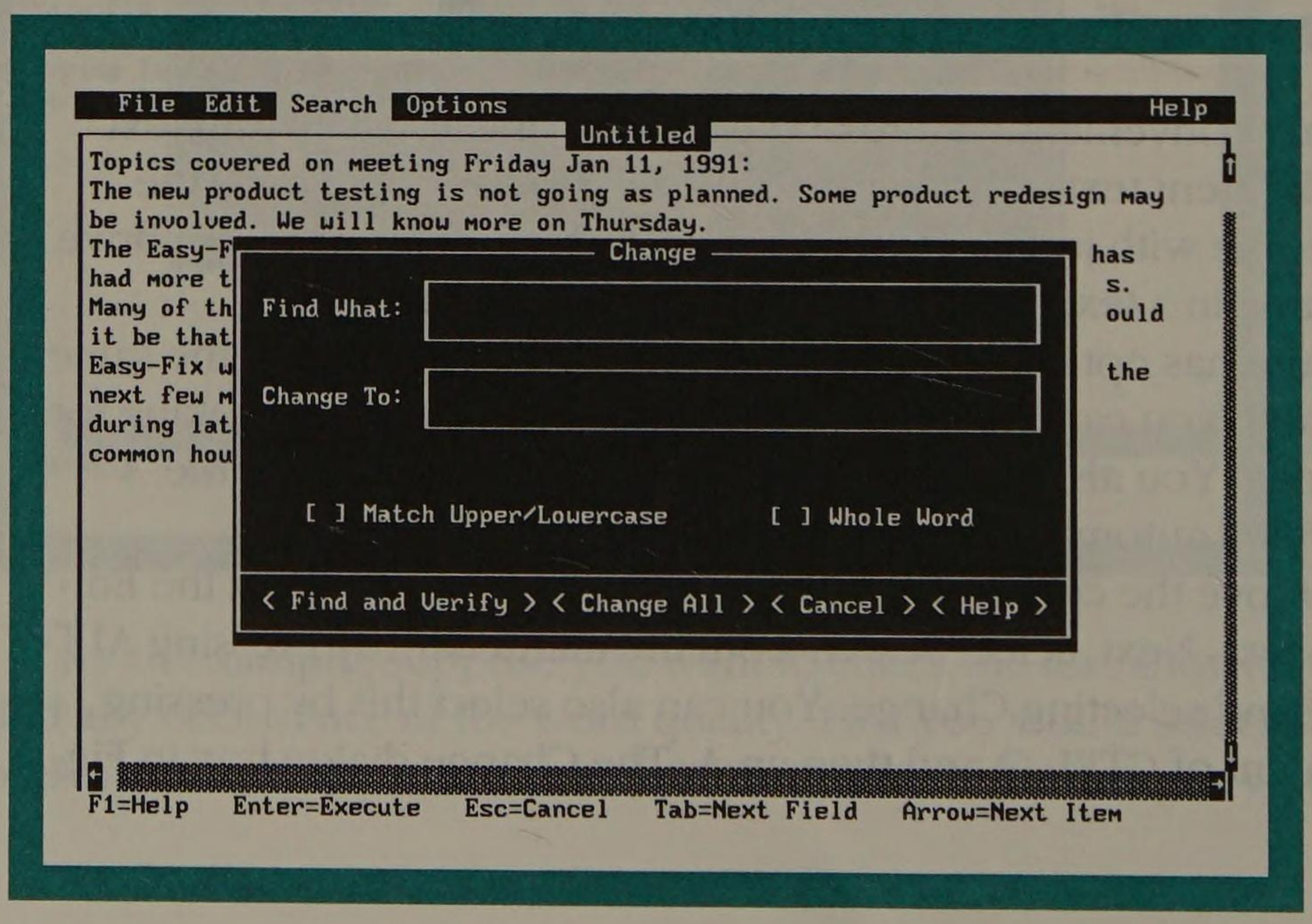
To replace text, move the cursor to the beginning of where you want the Editor to start replacing text. Next, select Search from the menu bar (by pressing ALT-S or clicking Search) and selecting Change. You can also select this by pressing the WordStar equivalent of CTRL-Q and then an A. The Change dialog box in Fig-

ure 11-6 appears. The Find What text box will be blank initially or will contain any selected text. You can type a new entry (which will replace the current one) or use the current entry the Editor displays. In the Change To box, you must supply the text that you want to use in place of the text in the Find What text box. You can also select the Match Upper/Lowercase and Whole Word check boxes to only find and replace words that have the same case or that are the entire word rather than part of it.

As an example, suppose you want to replace the word Easy-Fix with Nu-Glu in Figure 11-5. First you would select Search and Change or press the WordStar equivalent of CTRL-Q and an A. Next you would type **Easy-Fix** and press TAB or click on the Change To text box to move to the Change To text box. Type **Nu-Glu**. Press TAB twice and the space bar or click the Whole Word check box. Press ENTER or click on the Find and Verify button. When the Editor finds Easy-Fix, the Editor displays the dialog box shown in Figure 11-6. You can select Change to change this occurrence of Easy-Fix to Nu-Glu, select Skip to skip changing this occurrence of Easy-Fix and find the next one, or select Cancel to cancel the Search Change command.

Another option is to select Change All from the initial Change dialog box to change all occurrences of Easy-Fix to Nu-Glu without confirmation. When this command is finished, select OK or press ESC from the Change Complete dialog box. After the Editor goes through the document to the end, it starts again at the beginning.

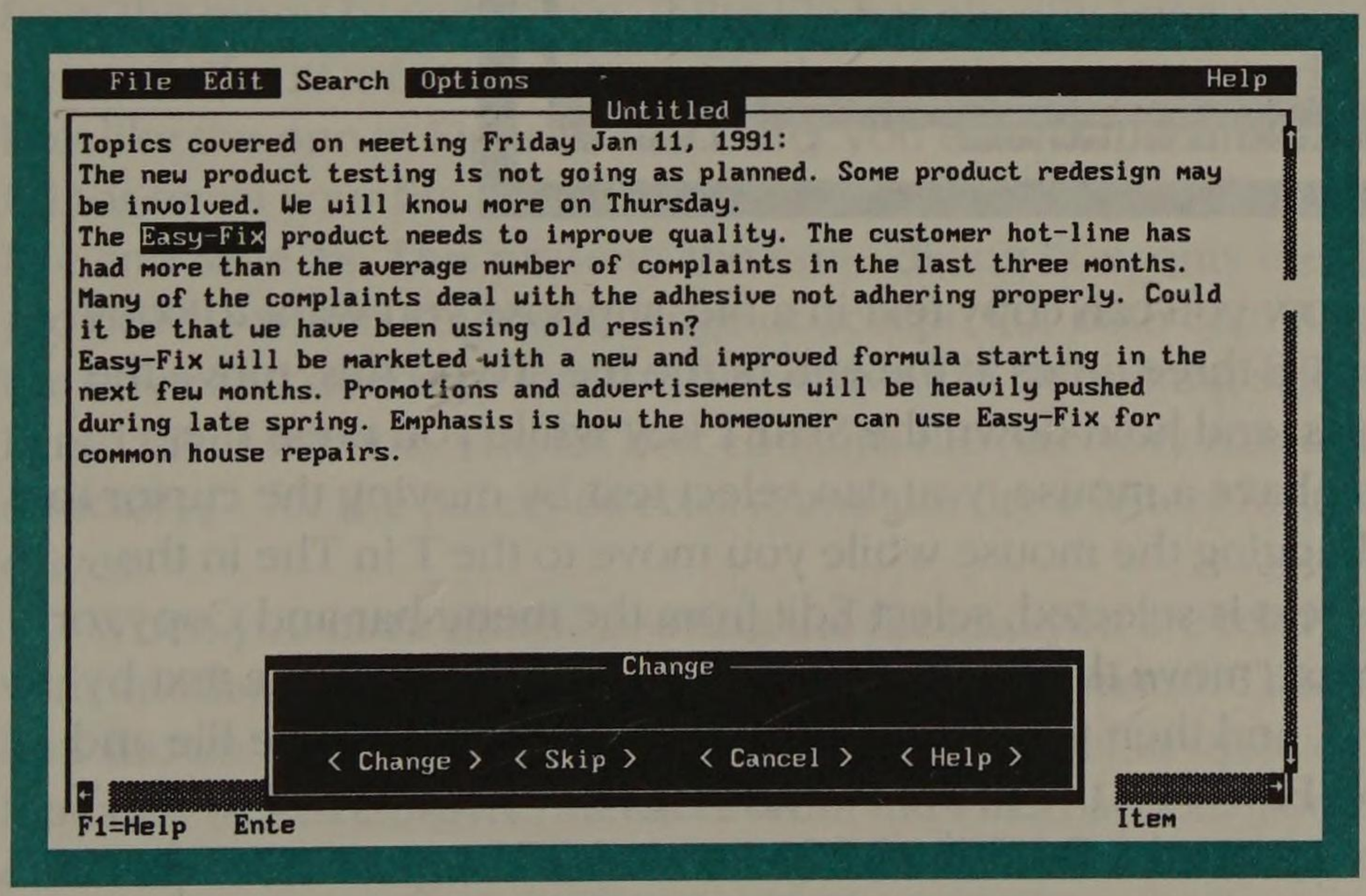
Figure 11-6: Change Dialog Box



Moving Text As you edit a text file you may want to rearrange the text in the file. You can rearrange text by moving it to a new location. Moving text requires that you select the text, cut it out of the file and into the Clipboard, move the cursor to where you want the text and copy it from the Clipboard to the file. The Clipboard is temporary storage that the Editor uses to store text that you copy and cut to it. Data remains in the Clipboard until you leave the Editor or until you put more data in the Clipboard that replaces the original Clipboard data. Cutting text to the Clipboard is done with the Edit Cut command and pasting text from the Clipboard is done with the Edit Paste command. These two commands have the shortcuts of SHIFT-DELETE and SHIFT-INSERT.

As an example of how you would move text in a file, suppose you wanted to move the second and third lines in Figure 11-7 to the end. To do this, you would move the cursor to the T in The and hold down the SHIFT key while you press the DOWN ARROW twice to move to the end of the text you want to move. If you are using a mouse, you can select this text by moving the cursor to the T in The and dragging the mouse to the T in The in the fourth line. Once this text is selected, select Edit from the menu bar and Cut, or press SHIFT-DELETE. Next, move the cursor to where you want the text added by pressing CTRL-Q and an R or by clicking below the c in common. Select Edit from the menu bar and Paste or press SHIFT-INSERT. Your text will now look like Figure 11-8.

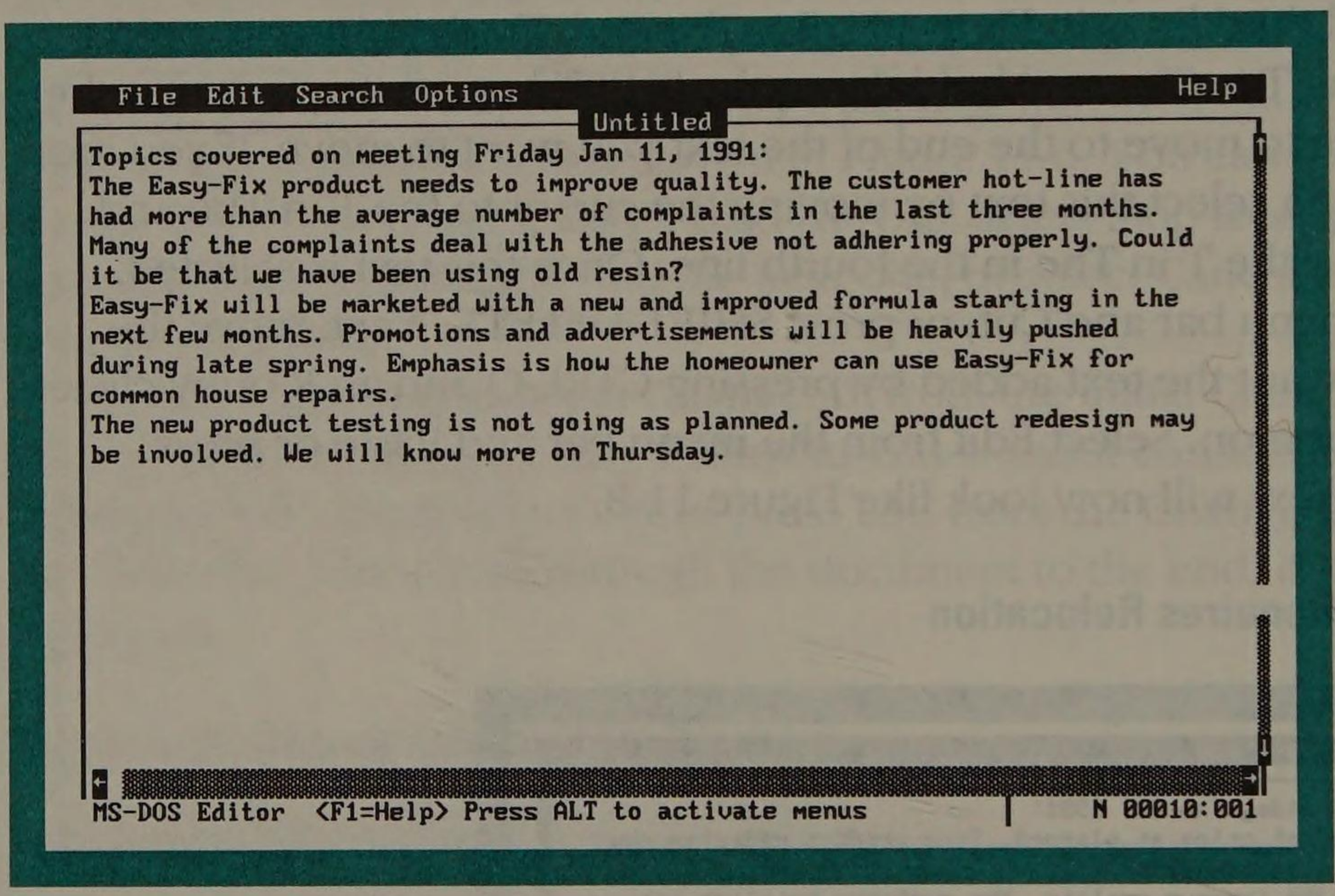
Figure 11-7: Text that Requires Relocation


File Edit Search Options Help
Untitled
Topics covered on meeting Friday Jan 11, 1991:
The new product testing is not going as planned. Some product redesign may be involved. We will know more on Thursday.
The Easy-Fix product needs to improve quality. The customer hot-line has had more than the average number of complaints in the last three months. Many of the complaints deal with the adhesive not adhering properly. Could it be that we have been using old resin?
Easy-Fix will be marketed with a new and improved formula starting in the next few months. Promotions and advertisements will be heavily pushed during late spring. Emphasis is how the homeowner can use Easy-Fix for common house repairs.
Change
< Change > < Skip > < Cancel > < Help >
F1=Help Ente
Item

Copying Text Another use of the Clipboard is for copying data. Copying text requires that you select the text, copy it into the Clipboard, move the cursor to

where you want a copy, and copy it from the Clipboard to the file. As described, the Clipboard is where the Editor temporarily stores text as you cut and copy text from the file. Copying text to the Clipboard is done with the Edit Copy command and pasting text from the Clipboard is done with the Edit Paste command. These two commands have the shortcuts of CTRL-INSERT and SHIFT-INSERT. You can also use these commands and steps to copy information from a help screen to a document by copying text from a help window. The Editor will not let you paste text into a help window.

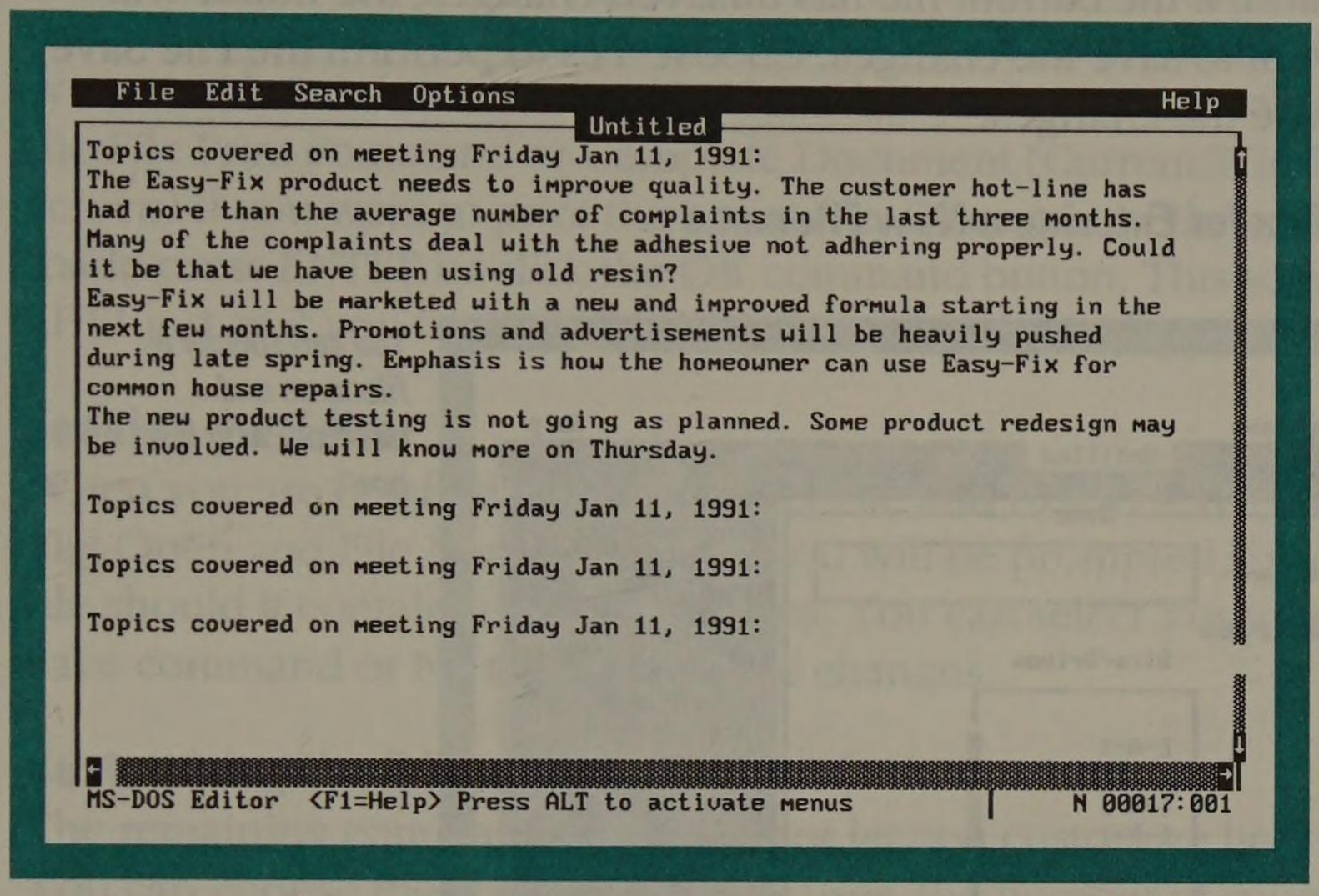
Figure 11-8: Text After Change Is Made


File Edit Search Options Help
Untitled
Topics covered on meeting Friday Jan 11, 1991:
The Easy-Fix product needs to improve quality. The customer hot-line has had more than the average number of complaints in the last three months. Many of the complaints deal with the adhesive not adhering properly. Could it be that we have been using old resin?
Easy-Fix will be marketed with a new and improved formula starting in the next few months. Promotions and advertisements will be heavily pushed during late spring. Emphasis is how the homeowner can use Easy-Fix for common house repairs.
The new product testing is not going as planned. Some product redesign may be involved. We will know more on Thursday.
MS-DOS Editor <F1=Help> Press ALT to activate menus
N 00010:001

The lines which were originally second and third are now at the bottom of the file.

As an example of how you can copy text in a file, suppose you want to copy the first line in Figure 11-8 three times at the end of the file. To do this, move the cursor to the T in Topics, and hold down the SHIFT key while you press the DOWN ARROW. If you have a mouse, you can select text by moving the cursor to the T in Topics, and dragging the mouse while you move to the T in The in the second line. Once this text is selected, select Edit from the menu bar and Copy, or press CTRL-INSERT. Next, move the cursor to where you want to add the text by pressing CTRL-Q and C, and then press ENTER to move to the end of the file and add a blank line. Select Edit from the menu bar and Paste or press SHIFT-INSERT. To make the next copy, press the DOWN ARROW and ENTER to move to where you want the next copy and press SHIFT-INSERT. After doing this again, your file will look like Figure 11-9.

Figure 11-9: Copy Operation Completed


File Edit Search Options Help
Untitled
Topics covered on meeting Friday Jan 11, 1991:
The Easy-Fix product needs to improve quality. The customer hot-line has had more than the average number of complaints in the last three months. Many of the complaints deal with the adhesive not adhering properly. Could it be that we have been using old resin?
Easy-Fix will be marketed with a new and improved formula starting in the next few months. Promotions and advertisements will be heavily pushed during late spring. Emphasis is how the homeowner can use Easy-Fix for common house repairs.
The new product testing is not going as planned. Some product redesign may be involved. We will know more on Thursday.
Topics covered on meeting Friday Jan 11, 1991:
Topics covered on meeting Friday Jan 11, 1991:
Topics covered on meeting Friday Jan 11, 1991:
MS-DOS Editor <F1=Help> Press ALT to activate menus
N 00017:001

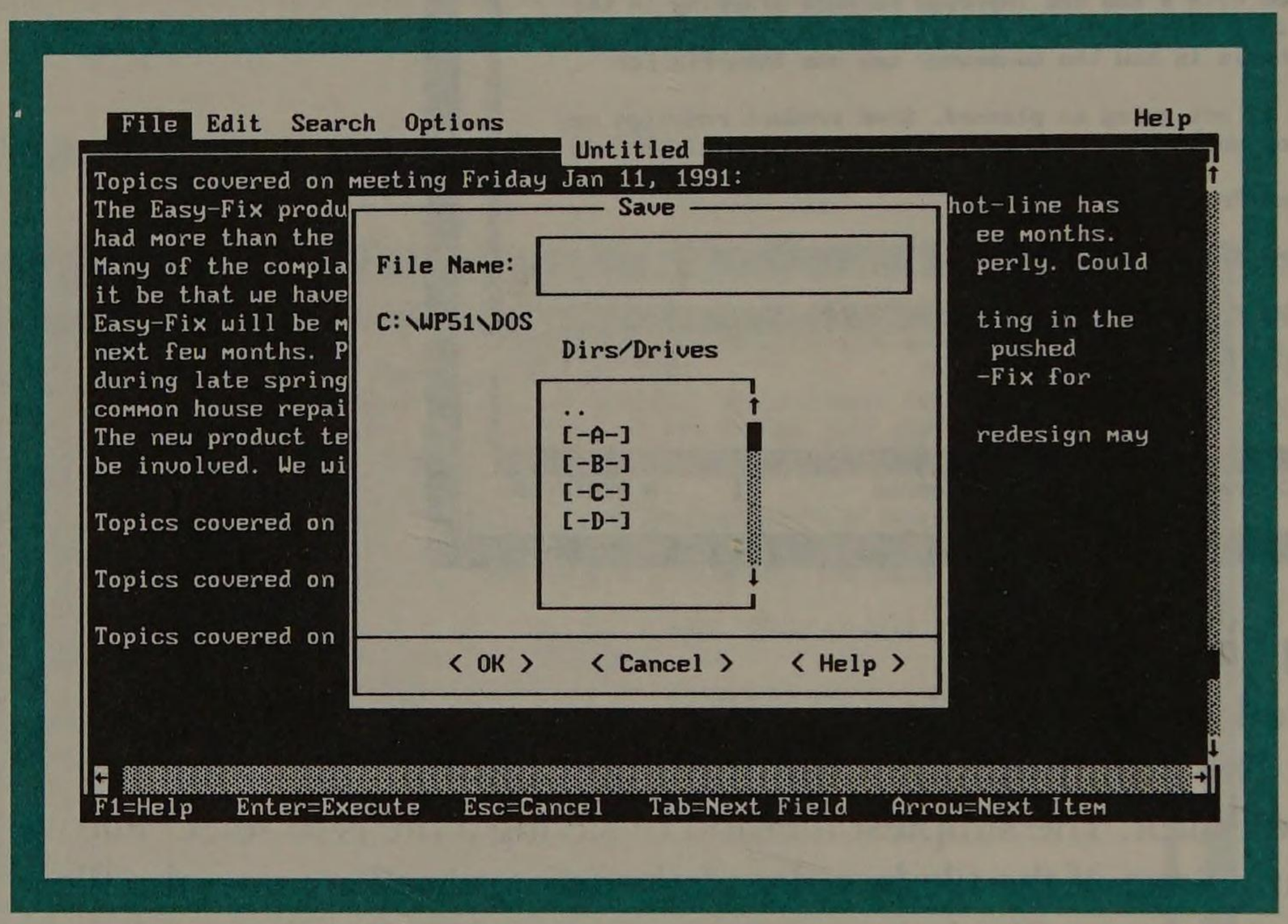
Saving and Printing Your Text File

When you are finished editing your file, you will want to save your changes and maybe print the file. The File menu includes several commands to save your file, print it, and leave the Editor. The simplest method of saving a file is to select File from the menu bar and Save. If the file has already been provided a name, the file is saved. If you are using a new file that you have not named, you will see a dialog box like the one in Figure 11-10. After you enter a filename and press ENTER, the Editor saves your file. You can also save a file using a different name by selecting File and Save As. This presents the same dialog box, only the filename is provided—which you can change or accept. For both commands, you can select the drive and directory that the file is saved to by making selections in the Dirs/Drives list box. The list box contains the directory names in the current directory, .. for the parent directory, and the drive letters for the other available drives.

When you have finished saving the file and you are ready to start working with another one, you can use the File Open command. This presents the dialog box shown in Figure 11-11. You must select the file to open by typing the filename or selecting the filename from the Files list box. You can change the directory containing the files listed in the Files box by making selections in the list box. The list box contains the directory names in the current directory, .. for the parent directory, and the drive letters for the other available drives. Another option

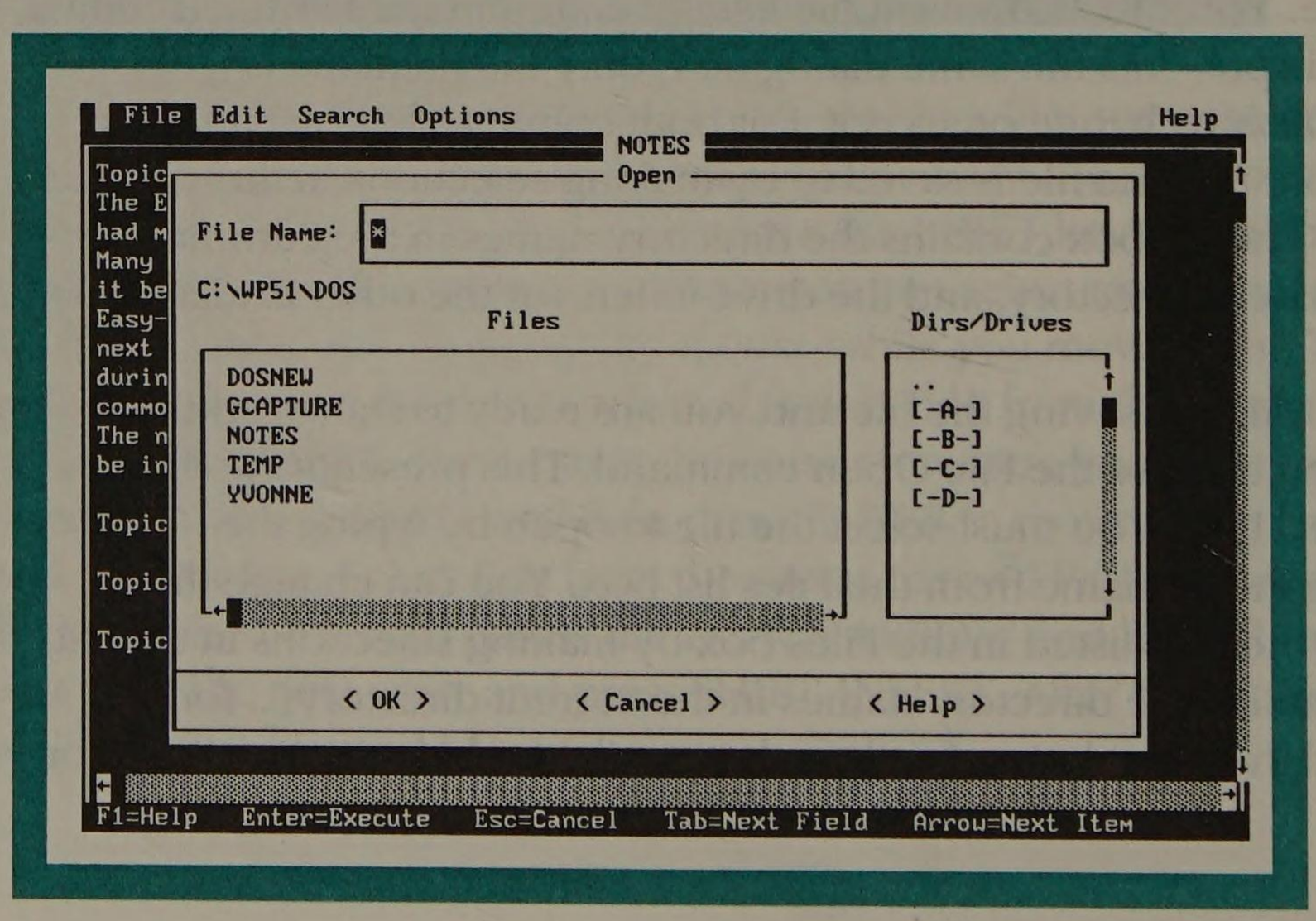
is to start with an empty file by selecting File and New. With both the File Open and File New command, if the current file has unsaved changes, the Editor will prompt you if you want to save the changes. Choose Yes to perform the File Save command or No to lose the changes.

Figure 11-10: Dialog Box for Entering a New Filename



You can specify a filename and directory with this box.

Figure 11-11: Dialog Box for File Open



You can either type a filename or select one.

You can print the current file or a portion of it. You can also print part of a help window screen or all of it. If you only want to print selected text, select the text you want to print. Next select File and Print. In the Print dialog box, you will want to select Selected Text only if you have selected text to print before selecting the File Print command or Complete Document (Current Window for a help screen) if you want to print the entire file. Once one of these two selections is made press ENTER or click the OK command button. This sends the output to LPT1 just as if you had used the COPY command to copy a file to LPT1.

Leaving the Editor

When you are finished, you can select File and Exit to leave the Editor. Like the File Open and File New command, you will be prompted if you want to save the file should it contain unsaved changes. You can select Yes to perform the File Save command or No to abandon the changes.

Customizing the Editor

The remaining commands in the Editor let you customize how the Editor works. You can choose the colors the Editor uses, the number of spaces each TAB uses, whether scroll bars are displayed, and where the file used by help is stored.

To display and change most of these settings, select Options from the menu bar and Display. The Display dialog box displays the current selections for the foreground and background colors, the check box for displaying scroll bars, and the current tab stop setting (how many spaces apart each tab stop is placed to allow you to press TAB and have the cursor move one tab stop to the right).

Changing Screen Colors For the colors, you can click the color you want or use the UP ARROW and DOWN ARROW to select the color you want. As you change the color, the dialog box shows how the color combinations will appear.

Removing the Scroll Bars and Changing the TAB Stops With the Scroll Bars check box, you can deselect the check box to remove the scroll bars—you may want to do this if you are not using a mouse. For the Tab Stops, you can enter the number of spaces represented by each tab stop. The tab stop setting is only used by the Editor. When you view the file in another editor or with the TYPE command, the file uses the default tab stops of eight spaces. You may want to change it to five if you are using the Editor to compose brief memos and you are pressing TAB to start each paragraph. Changing the tab stop only changes the number of spaces TAB represents for the times after you change the setting.

Using Other Customization Options The remaining customization option selects where the file EDIT.HLP is stored. On a hard disk, the files the Editor uses are in the DOS subdirectory. By including \DOS in a PATH statement, you do not need to tell the Editor where it can find the EDIT.HLP file. If you are not using a PATH statement or you are using floppy disks, you may need to tell the Editor where it can find the EDIT.HLP file. To do this, select Options and Help Path, then type the path information.

At a Glance

In this chapter you have learned how to start EDLIN, the text editor supplied as the only option for prior releases of DOS. You have also been introduced to the System Editor introduced with DOS 5 as a replacement for the EDLIN. Some of the specific tasks you have learned to handle are:

- **Start EDLIN at the DOS prompt—Type EDLIN**
- **Start the System Editor—Type EDIT at the DOS Prompt or select MS-DOS Editor from the Main Program Group in the DOS Shell**
- **Add a marker to the text file—CTRL-K followed by 0-3**
- **Locate a marker—CTRL-Q followed by 0 - 3**
- **Search for text—ALT-S Search or CTRL-Q F**
- **Replace text—ALT-S Change or CTRL-Q A**
- **Move selected text—SHIFT-DELETE, move cursor, SHIFT-INSERT**
- **Copy selected text—CTRL-INSERT, move cursor, SHIFT-INSERT**
- **Save the file—Select File Save from the menu**
- **Exit the editor—Select File Exit**
- **Customize the System Editor—Options Display**

Chapter 12

Defining a Series of Batch File Commands

Batch files are text files that contain a series of DOS commands. You can create these files with the System Editor, the DOS EDLIN program, or a word processor package that allows you to write out the data you enter as an ASCII file. Although you may currently think of batch files as programs beyond your capabilities, you will see that if you follow a few rules batch files can be created by anyone that uses DOS. The procedure is not complicated and you can limit the commands in batch files to the ones you already know how to enter at the DOS prompt.

What You'll Learn

After reading this chapter you will be able to:

- **Explain what a batch file is**
- **State how a batch file can save you time**
- **Use DOS commands in batch files**
- **Display messages on the screen**
- **Wait for a user response during batch file execution**
- **Document the commands in batch files**
- **Describe the purpose of AUTOEXEC.BAT**

Understanding What a Batch File Is and What it Offers

Batch files are text files that contain DOS commands. Batch files must be text files because DOS cannot process characters other than the ones you type at your keyboard. The instructions in batch files are identical to the commands you enter at the DOS prompt. These files are called batch files because the DOS commands in them are not entered individually, but are grouped and processed as a batch.

The only special requirement for batch files is that you use a filename extension of .BAT since this is what distinguishes them from other files and allows DOS to execute the instructions they contain.

Batch files offer an advantage over entries at the DOS prompt since they can be stored and reused whenever you need to perform the same set of tasks with DOS. You can use batch files to save yourself time when repetitive procedures are used. You can also use them to provide DOS capabilities to users who are not familiar with DOS commands. Once the DOS commands are entered in the batch file, all a user needs to do is type the name of the file (without the .BAT extension) to execute it. Having a selection of batch files is like having an assistant that can respond to your one-word commands and carry out tasks that would require a significant amount of typing. Once you have learned the procedure and created a few successful batch files, you will use them to handle all your repetitive DOS tasks.

Entering DOS Commands in Batch Files

You can choose several different methods for creating batch files of DOS commands. For a quick file with only a few lines, you can use the COPY command to create the file. Since this approach does not support changes you will want to use the System Editor, EDLIN, or a word processor for lengthier files. You can also use the DOSKEY history and save it as a file. Regardless of the approach that you use to enter the commands into the file you must follow the following rules for naming the file:

1. The filename must be a valid DOS filename from one to eight characters and it must have an extension of .BAT. The same restrictions on the use of special symbols and DOS reserved words that were discussed in Chapter 2 are also in effect for batch files.
2. Batch files should not have the same name as any of the internal DOS commands. The list of names you should avoid includes internal DOS commands, internal batch file commands not as yet discussed, and TRUENAME, an undocumented DOS command. The filenames you should avoid are: BREAK, CALL, CH, CHCP, CHDIR, CLS, COPY, CTTY, DEL, DATE, DIR, ECHO, ERASE, EXIT, FOR, GOTO, MD, MKDIR, PATH, PAUSE, PROMPT, RD, REM, RENAME, RMDIR, SET, TIME, TRUENAME, TYPE, VER, and VOL. If you store your .BAT file in the DOS directory and attempt to execute it, the DOS command with the same name will be executed instead.

3. Batch files should not be given the same name as program files (that is, files with a .COM or .EXE extension). Since these files are always executed first, you will not obtain the expected results.

Using COPY CON to Create a Quick Batch File

If you need a short batch file the quickest way to create it is with the COPY command. You tell DOS to copy everything that you type to a file by indicating that the source for the copy operation is the keyboard (this is indicated by the device name CON). To tell DOS to copy all that you type to a file called TRY.BAT, the format of the COPY command would be:

```
COPY CON TRY.BAT
```

When you press ENTER to finalize this command entry, the DOS prompt does not reappear. DOS is waiting for you to type information at the keyboard. When you finish each line of the batch file, you should check it and then press ENTER to have the line saved to the file. After pressing ENTER there is no way to go back and change the previous lines so you do not want to skip the checking step before finalizing. You will be on the next line and again a DOS prompt will not appear. You can continue to enter as many lines as you wish, pressing ENTER after each.

When you are finished entering lines in the batch file you need to inform DOS that you are finished and DOS will close the file. You can tell DOS that you do not have any more lines to add to the file by pressing F6 or CTRL-Z. Either approach will generate a ^Z on the current line. When you press ENTER, a message indicating that one file has been copied displays on the screen and the DOS prompt displays again. Figure 12-1 shows the entry of a few commands into a batch file using the COPY CON approach.

Figure 12-1: Commands Entered into a Batch File with COPY CON

```
C:\DOS>COPY CON QUICK.BAT
COPY *.BAT A:
COPY *.BAK A:
^Z
        1 file(s) copied

C:\DOS>
```

COPY CON entries are only suitable for a few short entries such as these.

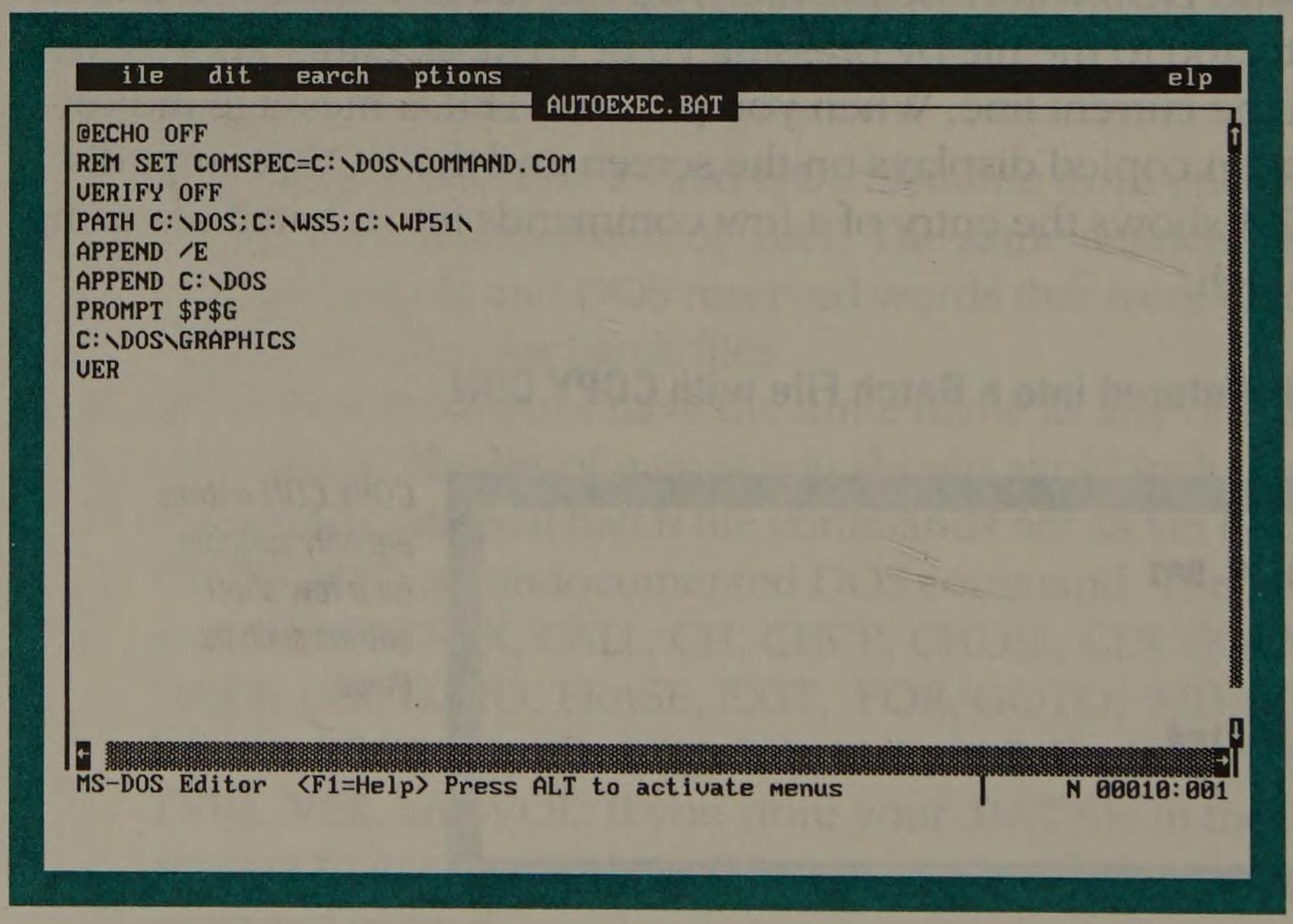
You can replace this file with another COPY CON operation but you cannot modify it with that approach. To modify the batch file you must use either the System Editor or your word processor since COPY CON does not support modifications.

■ ***NOTE:*** Regardless of which method you use when you originally create a batch file, you can use the System Editor, your word processor, or EDLIN when you want to revise it.

Using the System Editor to Enter and Save Batch Files

You learned to enter text with the System Editor in Chapter 11. It is ideal for entering and editing batch files as well as CONFIG.SYS since you can use its full screen capabilities when editing. If you have worked with its predecessor, EDLIN, and only have been able to work with one line of the file at a time, you will appreciate being able to move your cursor through the file, making changes at will. Although EDLIN is still available when you work at the DOS prompt (by typing **EDLIN**) you should make the required investment to convert to the newer full screen approach. Figure 12-2 shows the System Editor screen with lines entered for a batch file.

Figure 12-2: System Editor Screen with DOS Instructions Entered



DOS instructions are entered in the editor just as you would type them at the DOS prompt.

The System Editor was covered in full in Chapter 11. A brief summary of the EDLIN commands is also included in Chapter 11.

Using Your Word Processor to Create Batch Files

Word processing packages create special files when you use them to enter a document. These files contain the word processor's codes for special text features such as boldface and underlining. DOS does not support this type of file and you cannot create a document and expect to execute the code that it contains from DOS, even if you assign a .BAT filename extension to the file.

To create a DOS batch file with your word processor you will need to find out what special actions you need to take to write the file to disk as an ASCII text file without the special word processing codes. With some word processors, the special action to take before you start entering any data into the file (and with other processors after completing your entries) is that you must request that the file be written to disk in a special manner. You will want to check the following topics in your word processing manual to find out the approach used for writing files without the special codes. Topics that you might check in the index of your documentation manual are: ASCII text files, DOS text, and text files.

Figure 12-3 shows a batch file entered on a WordPerfect screen. The only special action needed with WordPerfect is to use the CTRL-F5, DOS Text, and Save options to write the batch file instructions to disk.

Figure 12-3: Batch File on a WordPerfect Screen

```
File Edit Search Layout Mark Tools Font Graphics Help
echo off
cls
echo.
echo.
echo                    ABC Company Software Selections
echo.
echo          A. 1-2-3 Release 3          F. WordPerfect 5.1
echo.
echo          B. Lotus Agenda             G. Quattro Pro
echo.
echo          C. Symphony                 H. WordStar 5.5
echo.
echo          D. dBASE IV                 I. Quicken 3
echo.
echo          E. Harvard Graphics         J. Microsoft
Project
echo.
echo.
echo       Type letter representing your selection and press
Enter.
echo.

C:\MENU.BAT                                   Doc 1 Pg 1 Ln 1" Pos 1"
```

The batch file will display a menu on your screen. You would need to create a series of batch files to process the user's menu selections.

Executing a Batch File

You can execute a batch file from the DOS prompt or from the DOS Shell. The results will be the same with either approach and the best choice will depend on where you are working at the time you want to perform the commands stored in the batch file.

Once you create a batch file you can execute the DOS commands that the file contains by typing the filename without the extension at the DOS prompt. In the case of the TRY.BAT file discussed earlier, you would type **TRY** to have DOS execute the commands in the file.

When you type an entry at the DOS prompt, DOS first checks your entry to see if it is an internal DOS command. If it is not, DOS checks the current directory to see if the entry matches the filename of a file with a .COM or .EXE extension. If there is a matching .COM or .EXE file, DOS executes the instructions in the file. If not, DOS searches the current directory for a file with a .BAT extension and if it finds one it executes the DOS commands it contains. If DOS does not find a match for the filename you entered, it will also check the directories listed in the most recent path statement. Figure 12-4 provides a graphic illustration of the checking that DOS does when you try to execute a batch file.

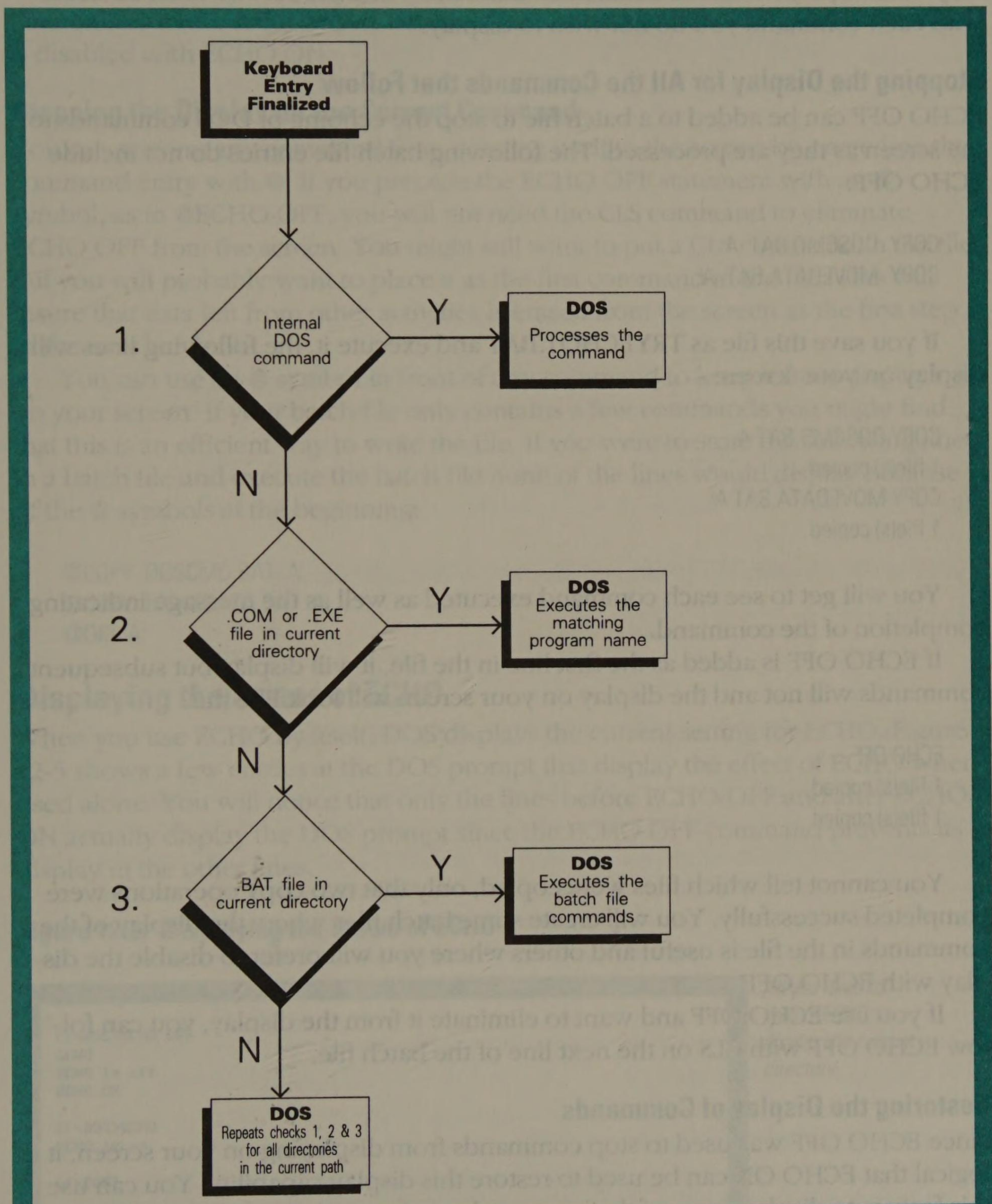
You can execute a batch file from the DOS Shell by activating the File menu with ALT-F or a mouse click. You can type **R** for Run or click on the RUN command. After you type the name of your batch file in the Run box, DOS will run the batch file and display your results. You can press any key at the end to return to the DOS Shell. Another approach from the Shell is to highlight the name of the batch file that you wish to execute from the file list, next select File, then choose Open by typing **O** or Clicking on the Open option. DOS executes the commands in the highlighted filename and returns you to the Shell window when you are through.

You can also execute one batch file from within another batch file by entering the name of the batch file you want to execute on a line in the batch file that you plan to execute first.

Controlling the Display of Commands

When DOS executes the commands in a batch file, it displays the command as it is entered in the batch file and then any output from the command beneath the command. The display of the command being processed can distract from the display of the command result or it can enhance the information provided. For a user who is not familiar with DOS, the display may be confusing since they will not be familiar with DOS command entries.

Figure 12-4 : Checks DOS Performs When You Complete an Entry on the Command Line



You can stop the display of commands from the screen in two ways. One option stops the display for all the commands that follow and the second must be used with each command you do not wish to display.

Stopping the Display for All the Commands that Follow

ECHO OFF can be added to a batch file to stop the echoing of DOS commands to the screen as they are processed. The following batch file entries do not include ECHO OFF:

```
COPY DOSCMD.BAT A:
COPY MOVEDATA.BAT A:
```

If you save this file as TRYECHO1.BAT and execute it, the following lines will display on your screen:

```
COPY DOSCMD.BAT A:
1 file(s) copied
COPY MOVEDATA.BAT A:
1 file(s) copied
```

You will get to see each command executed as well as the message indicating completion of the command.

If ECHO OFF is added as the first line in the file, it will display but subsequent commands will not and the display on your screen will look like this:

```
ECHO OFF
1 file(s) copied
1 file(s) copied
```

You cannot tell which files were copied, only that two copy operations were completed successfully. You will create some batch files where the display of the commands in the file is useful and others where you will prefer to disable the display with ECHO OFF

If you use ECHO OFF and want to eliminate it from the display, you can follow ECHO OFF with CLS on the next line of the batch file.

Restoring the Display of Commands

Since ECHO OFF was used to stop commands from displaying on your screen, it is logical that ECHO ON can be used to restore this display capability. You can use this feature to display some of the commands on the screen and not display others.

If you enter ECHO OFF at the DOS prompt, the DOS prompt will no longer display with each new line on your screen although DOS will continue to process your entries in the same way as before. You can restore the DOS prompt when it is disabled with ECHO ON.

Stopping the Display for the Current Command

You can prevent any command from echoing itself to the screen by preceding the command entry with @. If you precede the ECHO OFF statement with an @ symbol, as in @ECHO OFF, you will not need the CLS command to eliminate ECHO OFF from the screen. You might still want to put a CLS command in the file but you will probably want to place it as the first command in the file. This will insure that data left from other activities is erased from the screen as the first step in the new batch file.

You can use an @ symbol in front of any command to keep it from displaying on your screen. If your batch file only contains a few commands you might find that this is an efficient way to write the file. If you were to store the following lines in a batch file and execute the batch file none of the lines would display because of the @ symbols at the beginning:

```
@COPY DOSCMD.BAT A:
@COPY MOVEDATA.BAT A:
@DIR A:
```

Displaying the Status of ECHO

When you use ECHO by itself, DOS displays the current setting for ECHO. Figure 12-5 shows a few entries at the DOS prompt that display the effect of ECHO when used alone. You will notice that only the lines before ECHO OFF and after ECHO ON actually display the DOS prompt since the ECHO OFF command prevents its display in the other lines.

Figure 12-5: Displaying the Status of ECHO

```
C:\DOS>ECHO OFF
ECHO
ECHO is off
ECHO ON

C:\DOS>ECHO
ECHO is on

C:\DOS>
```

All you need to type is ECHO to display the current directory.

Using ECHO to Display Other Information

Once you turn the echoing feature off, you can use the ECHO command to your advantage in displaying all types of information on the screen. You might display a message to users asking them to insert a data disk in a drive before starting a program or display a list of menu options from which they can choose. The format of the command is ECHO followed by the message you want to display. The following lines show how the ECHO commands might be entered in a batch file:

```
ECHO OFF
CLS
ECHO Backup procedure is about to start
ECHO You will need to place a formatted disk in drive A
```

Executing this batch file would produce the display shown in Figure 12-6. You will notice that the CLS command erases the screen and causes the two messages displayed with ECHO to appear at the top of the screen.

Figure 12-6: Messages Placed at the Top of the Screen with ECHO Commands

```
Backup procedure is about to start
You will need to place a formatted disk in drive A
C:\DOS>
```

The command ECHO does not display, you will only see the message here.

Adding Blank Lines

It is sometimes easier to read information when there is extra space around it. You can add space between the lines created by ECHO. To send a blank separator line to your screen all you need to type is **ECHO.**. The period tells DOS that a blank line should be inserted. You will find the blank lines especially useful in a later example where you want to display a whole series of menu options on the screen, yet produce an attractive display.

■ ***NOTE:*** Extra blank lines improve readability and lower errors when you are providing users a list of options

Interrupting Batch File Execution

You can plan for a pause in the action of a batch file by inserting an instruction which will cause the file to wait for the user to indicate that it is appropriate to

continue. You can also either permanently or temporarily interrupt batch file execution from the keyboard.

Adding a Command to a Batch File that Causes Execution to Pause

If you need a user to check the label on a disk or insert a new disk in the drive before executing the next command, you will not be able to use keyboard intervention to stop execution of the file. You must plan for this pause in execution flow and place a PAUSE command in the batch file at the location where you want the pause to occur. PAUSE is frequently preceded by an ECHO statement to inform the user of the reason for the pause. You might enter these instructions in the batch file:

```
ECHO  Place a blank disk in Drive B
PAUSE
COPY  *.*  B:
```

If ECHO is OFF, DOS display the message from the ECHO command followed by instructions for continuing after PAUSE:

```
Place a blank disk in Drive B
Press any key to continue . . .
```

After a key is pressed indicating that a disk has been inserted, the COPY instruction is executed and uses the newly inserted disk as the target location for the files that are being copied.

Temporarily Stopping Execution from the Keyboard

If you press the CTRL-S key sequence, DOS will temporarily suspend execution of the batch file that is currently running. To resume execution where you left off, you can press CTRL-S again. This quick interruption is useful if you realize you need to insert a disk in drive A and want a few minutes to look for it. With this approach you can add the file before DOS displays an error message telling you it cannot find the file. You cannot use DOS prompt entries to complete other tasks while the batch file is paused.

Canceling Batch File Execution

If you realize that you have made a mistake in your request to execute a batch file you can press CTRL-C to stop the execution of the commands in the file. You cannot restart the execution where you left off, but must return to the beginning of the file if you wish to re-execute it later.

DOS at Work

(Heidi Thompson is President of H.P. Thompson & Co. Her company specializes in PC training and systems development.)

In my business I often need to copy all of the files from one floppy to 10 or 12 other floppies and so I developed the following batch file:

```
echo off
rem MAKECOPY.BAT is name of file
rem allows to copy all files on one floppy to other floppies
cls
echo on
cd\copy
del\copy\*.*
echo off
cls
echo Put original in A: and press Enter
pause
copy a:*.*
:next
echo Change disks in A: and press enter or CTRL-BREAK TO STOP!!
pause
copy *.* a:
goto next
rem END OF FILE
```

To begin I type ***MAKECOPY*** *at the DOS prompt. The file then changes to the \COPY directory and deletes the files. At this point I must type* **Y** *to confirm. It then prompts me to put the disk I want to copy in A: and all files are copied from A: to the hard drive. It then prompts me to change disks and pause until I press ENTER. The files are copied back to A: and the program again pauses for me to switch disks and press ENTER. It returns to the statement ":next" and continues. Once I have made as many copies as I need, I simply press CTRL-BREAK at the pause and when prompted "Terminate batch job (Y/N)?" type* **Y**. *When the program terminates, I am in the \COPY directory with all of my copied disks in hand.*

Documenting Batch Files

At the time you create a new batch file, it seems perfectly clear what tasks the batch file completes and your reasons for creating it. One month, six months, or a year from now, the purpose of the batch file may not seem at all clear. You can add remarks at any location within the file should you wish to document the overall purpose of the file or of a specific command within the file. The remark statements will not affect the execution of a batch file since they are not processed and do not display during execution. They will display if you print the file from EDLIN or the System Editor or with the DOS PRINT command.

To add a comment to a batch file you use the REM command followed by any text you want to use. You can use REM commands at any location with a batch file and are not limited in the number that you use. A single REM statement cannot exceed the DOS limit of 256 characters. The following lines might be used as remark statements within a batch file:

```
REM  MENU.BAT
REM  This batch file displays the application menu for Harris, Inc.
```

You can continue a REM entry beyond one line as you type, although it is often clearer to add a REM statement at the beginning of each new line.

■ ***NOTE:*** The length limitation is 253 characters when you are working with EDLIN and 256 characters when you are entering DOS commands through the System Editor.

Testing Batch Files

Since the DOS commands are not processed by DOS as you enter them, the batch file you are creating may not work properly when you need to execute it. The entries in the file may contain syntax errors such as a misspelled command name or a switch that the command does not support. The batch file can also contain errors in logic.

Correcting Syntax Error Problems

If a command in a batch file has an incorrect syntax, DOS will display an error message and proceed to the next command. The error message will depend on what is wrong with the entry. The message that displays will be the same message you would see if you made the same syntax error with an entry at the DOS prompt. For example, if you misspelled a command name, DOS would display **Bad command or file name**. DOS skips the command in error and proceeds to

the next command entry. You will want to try any batch files that you create at the time of creation to insure that they do not contain syntax errors.

Correcting Logic Errors

Logic errors occur more frequently with the advanced batch files discussed in Chapter 13, but they can also occur in a simple batch file. Logic errors include errors in execution flow throughout a batch file as instructions are not performed in the correct sequence, or are incorrectly performed repetitively. Errors also include differences in the way the batch file operates versus the way the user expects the batch file to work.

You might want the batch file to start the 1-2-3 program. If the 1-2-3 program file is located in the 123 directory, the batch file will either need to change the directory to 123, alter the PATH statement to add the 123 directory to the locations checked for program files, indicate the path when the name of the file is specified, or assume that the user will activate the 123 directory before invoking the batch file. Making the assumption that the user will change the directory before invoking the file is not a good assumption and may result in failure for the batch file since it will not find the program requested. A foolproof design would include instructions that would not be dependent on the user remembering to change the directory or perform other tasks to make sure the batch file will work.

Testing will also identify logic errors in batch files. With the more sophisticated variety you might need to test several times to insure you have tested the file under all sets of conditions. You should not consider the creation of a batch file to be complete until you have tested it under the conditions in which it will be used.

Looking at a Special Batch File

Most batch files are invoked by typing the name of the batch file at the DOS prompt or selecting Run or Open from the DOS Shell. You can also create a special batch file to be executed when you boot your system. This batch file must be stored in the boot directory of the disk you boot your system from and it must be named AUTOEXEC.BAT. You can use this file to control the active directory after booting your system, to display a menu or message for the users, or to change the DOS prompt.

The Purpose of AUTOEXEC.BAT

AUTOEXEC.BAT contains a series of commands that perform a consistent set of tasks. They allow you to focus on the next step since you do not have to waste time trying to remember the exact set of steps needed to start DOS in a consistent fashion. Since the entry of these commands can be lengthy, AUTOEXEC.BAT also

saves a significant amount of time when you start DOS since the instructions are stored on the computer and processed more quickly than you could possibly type them.

Looking at a Sample AUTOEXEC.BAT File

You can put any type of DOS commands and batch file commands that you want into AUTOEXEC.BAT. The following lines represent the entries that might appear in an AUTOEXEC.BAT file:

```
@ECHO OFF
VERIFY OFF
PATH C:\DOS;C:\WS5;C:\WP51\
PROMPT $P$G
PRINT /D:LPT1
C:\DOS\DOSSHELL
```

The first entry, @ECHO OFF, stops DOS from echoing the current command as well as subsequent commands to the screen. VERIFY OFF stops DOS from verifying data written to a file to speed up disk operations. The PATH statement indicates all the directories that should be checked for program and batch files when DOS cannot find the file it is looking for in the current directory. The PROMPT command alters the DOS prompt to display both the current drive and path. The PRINT statement sets the print output device to LPT1. Because this statement is included in the batch file executed when DOS is booted, the user will not be prompted for the output device the first time that they use the PRINT command. The last statement in the file executes the file DOSSHELL.COM, which causes the DOS Shell to display on the screen.

Other commands in common use in a batch file are DATE and TIME to prompt a user without a clock in their system to enter the correct date and time at the beginning of each DOS session. The CD command is also used in many batch files to control the active directory. Often, other batch files such as the menu file mentioned later in this chapter, are invoked. Including the menu or other detailed procedure in a separate file keeps the AUTOEXEC.BAT file shorter.

- ***NOTE:*** Always make a backup copy of AUTOEXEC.BAT under another name before revising it. You can delete the changed version and rename the old version if you decide you do not like the changes.

Creating a Few Sample Batch Files

The best way to get started with batch files is to create a few. The ones in the following sections offer a utility to almost any one since they create a backup disk and display a menu for your selection. You will need to customize the backup file to match the directory names where your data is stored. The batch file which creates a menu of choices also needs to be customized to match the application programs you have on your system.

Create a Batch File to Backup Your Hard Disk

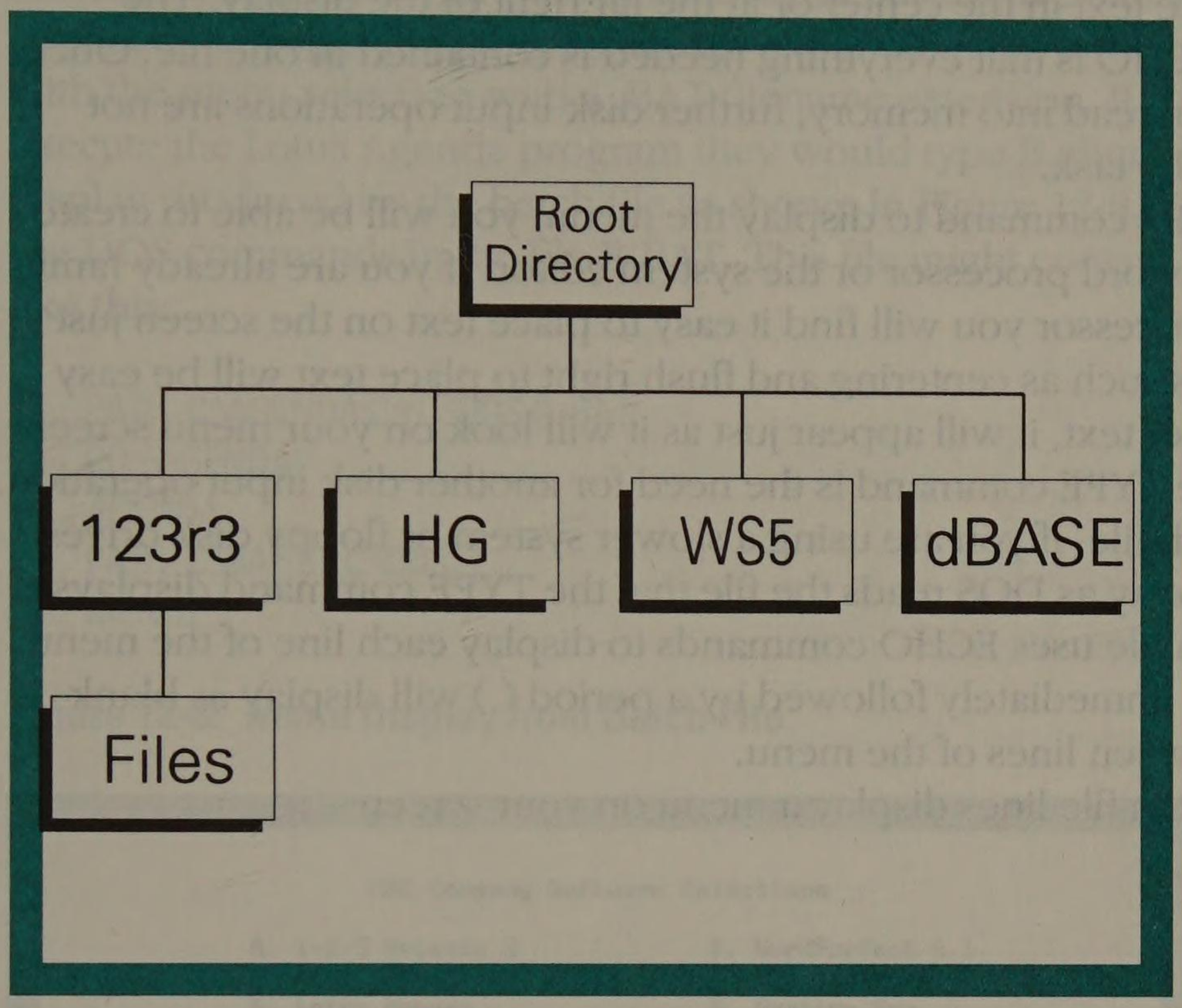
It is important to backup your hard disk frequently. Many users do not backup as often as they should. One reason that users are negligent about backup is that the most efficient way to backup involves backing up all the individual subdirectories and this procedure requires the entry of numerous DOS commands.

If the disk is backed up from the root all of the program data is unnecessarily copied to the hard disk each time you backup. If you backup from the individual data directories you do not need as many disks and the procedure does not take as long. The disadvantage of backing up from the lower level directories is that a DOS BACKUP command is needed for each directory you want to backup. If you are entering these instructions at the DOS prompt, you would need to wait until one directory was completed and then make the next request. Once you start the backup procedure, it is difficult to focus on other tasks because you need to constantly check to see if DOS needs the next command. A batch file can handle this difficult situation since you can store your directions once and invoke all of them with one command anytime you want to create a backup.

Given the directory structure in Figure 12-7 and data only in the lowest level directories you would need to backup four directories. If you want to start a new backup disk with the first directory that you backup and append the data for the other directories, your batch file code might look like this:

```
REM  BKUP
REM  This file contains instructions for creating a backup file
REM  The backup file will contain information from the subdirectories indicated.
BACKUP  C:\123R3\FILES\*.WK1  A:
BACKUP  C:\HG\*.CHT A:  /A
BACKUP  C:\WS5\*.DOC A:  /A
BACKUP C:\DBASE\*.CAT  A:  /A
```

Figure 12-7: Files to Back Up



Creating a Simple Menu System

On most computer systems, a hard disk makes it easy to organize a variety of application software by storing each application program in a separate directory. To start the application you typically change the directory to the one containing the desired application and type the name of the program. This means you must remember the name and spelling of each directory and the programs within them. Another alternative is to create a DOS batch file that presents a menu and allows you to invoke any application that is part of the menu system by typing a letter or number. This batch file can be used every time you work with the system and can even be shared with other users that are less familiar with DOS.

Creating the Menu Display There are two different ways to provide the menu display. You can use ECHO statements to display each line of the menu including blank lines. You can also store the menu as it will appear in a text file and use the TYPE command to display the contents of this file on your screen. Each approach has advantages and disadvantages.

If you choose to use ECHO to display the menu lines on your screen, you will need an ECHO statement for each line. Also, centering and text placement will be

difficult to judge since the ECHO command must precede each line and there are no commands to place text in the center or at the far right of the display. The advantage of using ECHO is that everything needed is contained in one file. Once the batch file has been read into memory, further disk input operations are not needed to complete the task.

If you use the TYPE command to display the menu, you will be able to create the menu using your word processor or the system editor. If you are already familiar with your word processor you will find it easy to place text on the screen just as you like it. Options such as centering and flush right to place text will be easy to invoke. As you enter text, it will appear just as it will look on your menu screen. A disadvantage of the TYPE command is the need for another disk input operation after reading the batch file. If you are using a slower system or floppy disk drives you may notice the delay as DOS reads the file that the TYPE command displays.

The sample batch file uses ECHO commands to display each line of the menu. The lines with ECHO immediately followed by a period (.) will display as blank line to add space between lines of the menu.

The following batch file lines display a menu on your screen:

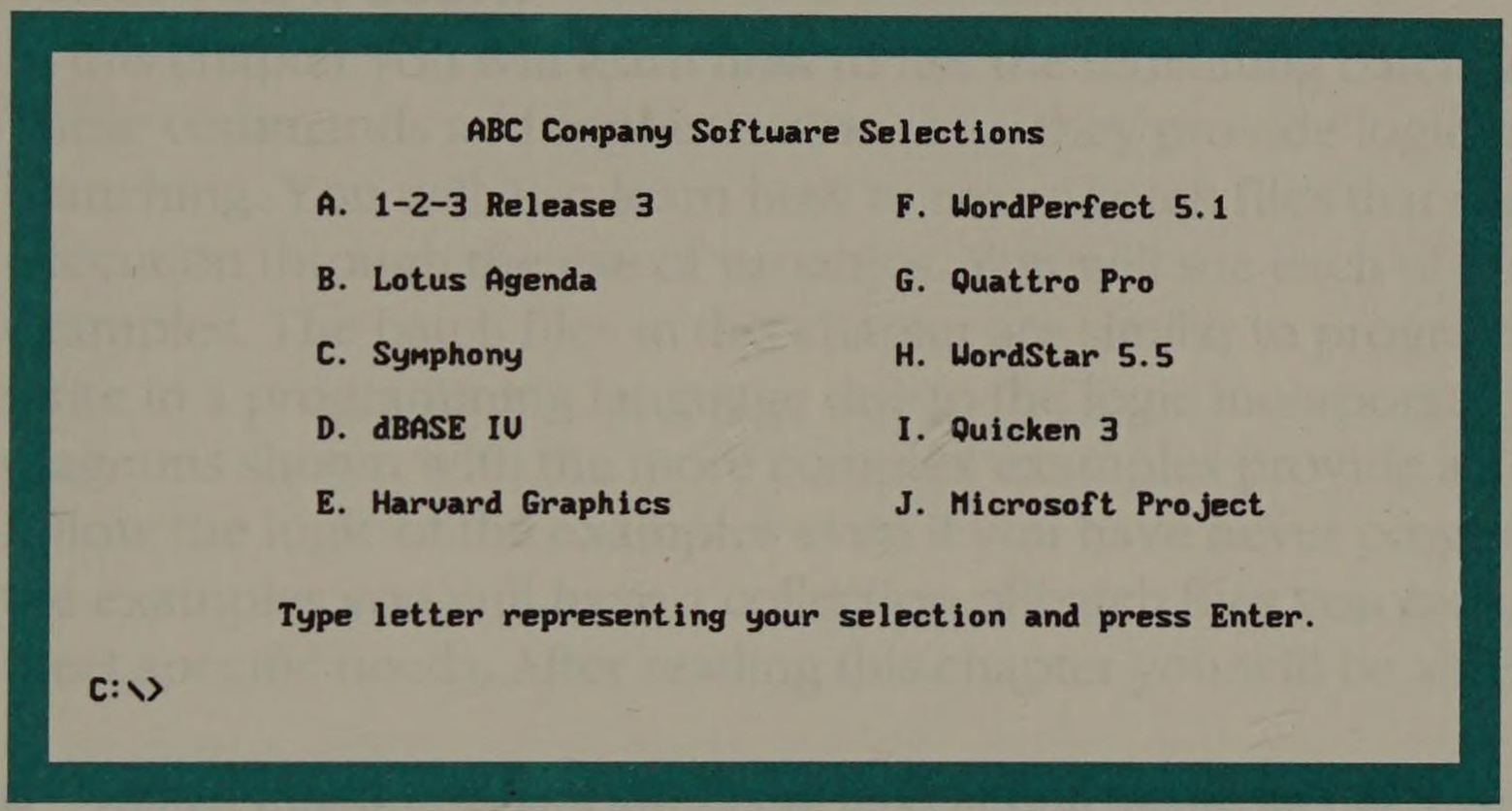
```
echo off
cls
echo.
echo.
echo.
echo          ABC Company Software Selections
echo.
echo      A. 1-2-3 Release 3          F. WordPerfect 5.1
echo.
echo      B. Lotus Agenda             G. Quattro Pro
echo.
echo      C. Symphony                 H. WordStar 5.5
echo.
echo      D. dBASE IV                 I. Quicken 3
echo.
echo      E. Harvard Graphics         J. Microsoft Project
echo.
echo.
echo      Type letter representing your selection and press Enter.
echo.
```

The extra blank lines added with ECHO., help to make this file more readable.

Providing a Way to Invoke Applications Each choice in your menu display must have a corresponding batch file that contains the necessary DOS instructions to begin the application program. The name of this batch file is the letter or number associated with the menu selection with a .BAT filename extension. If a user wanted to execute the Lotus Agenda program they would type B after viewing the menu display produced by the batch file as shown in Figure 12-8. DOS would execute the DOS commands in the file B.BAT. This file might contain instructions that look like this:

```
PATH C:\AGENDA;C:\;C:\DOS;C:\BAT;
CD \AGENDA
AGENDA
PATH C:\BAT;C:\;C:\DOS;
MENU2
```

Figure 12-8: Menu Display from Batch File



A menu like this makes it easy for a user to start his or her new task.

For the menu shown in Figure 12-8, you will need 10 batch files. The remaining files are similar to B.BAT, which is used when Agenda is selected. The differences will be in the modification to the PATH statement to add the directories that the application being invoked will use, the directory established as the current directory to start the application, and the name of the program used to invoke the new application. If the name of the batch file containing the menu displays is added as the last instruction, MENU2, the menu will display again after the user exits Agenda.

At a Glance

Batch files can increase your productivity with DOS. They are really quite easy to create if you remember that your objective is to record the commands you want DOS to execute. In this chapter you learned how to create your first batch file. Batch files can be as simple as a few DOS commands or they can contain special batch file commands that can only be used in a batch file. Some of the important points covered in this chapter are:

- **Create a quick batch file—COPY CON**
- **Create batch file with the Systems Editor—EDIT**
- **Document a batch file—REM**
- **Display a message on the screen—ECHO**
- **Wait for user to press a key—PAUSE**

Chapter 13

Adding Sophistication to Command Definitions

In Chapter 12 you learned how to create your own command definition by entering a series of DOS commands in a batch file. Most of the command entries in the sample files were the same commands that you enter at the DOS prompt. You were introduced to a few special batch file commands as you learned to display messages, temporarily stop a batch file, and document your entries with ECHO, PAUSE, and REM.

What You'll Learn

In this chapter you will learn how to use the remaining batch files commands. These commands add sophistication since they provide logic capabilities and branching. You will also learn how to create batch files that can change with each execution through the use of variables. You will see each of the new commands in examples. The batch files in this chapter are similar to programs that you might write in a programming language due to the logic incorporated within them. The diagrams shown with the more complex examples provide an easy way for you to follow the logic of the examples even if you have never programmed. If you enter the examples you will have a collection of batch files you can easily modify to meet specific needs. After reading this chapter you will be able to:

- **Check conditions and affect the next action based on the result of the test**
- **Change the sequence in which batch file instructions are executed**
- **Use the commands in multiple batch files with one entry**
- **Repeat a batch procedure**
- **Use variables in batch files**
- **Shift variable values**

Using Parameters in Your Batch Files

Most of the DOS commands you use work with parameters to allow you to use the command in many different situations. The batch files you have created up to this point are designed to perform the same task every time you use them. You can add parameters to your batch files that function like the parameters in the DOS commands with which you have become familiar. You can think of parameters as blanks you fill in when you are ready to execute the commands in a batch file. Just as you might construct sentences with blanks such as I had a _____ day at work and complete the blank with great, lousy, productive, or tiring, depending on what your day was like, you can leave the same types of blanks in batch files. Rather than including the names of files, directories, and disk drives in batch file commands, you can leave a place for the information to be supplied with a parameter at the time that you execute the batch file. The parameter values you supply at the time of execution will be plugged into the batch file at the places where you have included parameter place holders. Using this approach you can specify new information every time you use the batch file.

Representing Parameters

You can use parameters to represent any type of information you want. %0 through %9 are the 10 placeholders that are used to indicate a replaceable parameter. These placeholders can be used to mark the location where the name of a particular file, a directory, or a command will be supplied at execution. You can use a single placeholder or all 10 if you need them. You supply the value for the placeholder as a parameter when you execute the batch file.

Looking at Two Simple Examples Batch files that use parameters do not have to be complicated. The only requirement is that you want flexibility at the time of execution with the ability to control how the files operate at execution. You can change the files that are used with each execution or you can have the parameter represent the command that is executed.

Using a Parameter to Represent a File You might want to perform several tasks with one file such as copying the file and printing its contents. You can avoid having to type the filename twice if you use the replaceable parameter concept with a simple batch file. For each place that you want to reference the same filename, use the same placeholder. You can also change the file that you use each time you use the batch file. To create the batch file that performs both tasks you would make these entries:

```
COPY %1 A:
PRINT %1
```

If you save the file as BOTH.BAT you can execute it for the file BUDGET.TXT with the entry:

```
BOTH BUDGET.TXT
```

DOS substitutes the entry you have provided as the parameter value, BUDGET.TEXT, for both occurrences of %1 with the following result:

```
COPY BUDGET TXT A:
PRINT BUDGET.TXT
```

The commands execute as if the entries BUDGET.TXT had been typed in the batch file at the time that you originally recorded it.

Using a Parameter to Represent Commands If you sometimes want to display the contents of a file on the screen and at other times you want to print the contents of the file on your printer, you can use a parameter to represent the command that you want to execute. You can supply the command at the time of execution as a parameter value. If the batch file always copies the file to a disk in drive A, it might look something like this:

```
COPY MASTER A:
%1 MASTER
```

If the batch file is called MASTER.BAT you would enter MASTER PRINT to print the file MASTER. The second line of the batch file would have PRINT substituted for %1. You would type MASTER TYPE to display the file on the screen. The second line in the batch file executes as if you had entered TYPE MASTER instead of %1 MASTER.

Order of Parameter Assignment to Placeholder The placeholders %0 through %9 are assigned sequentially to each piece of information entered to invoke a batch file. You may feel that you should have used %0 instead of %1 to represent the first piece of variable data when you entered the previous examples. Remember that the name of the batch file is the first piece of information that can vary. The assignment of %0 to this batch filename enables you to display the name of the

batch file that is currently executing with an ECHO statement that contains the %0. An entry such as ECHO %0 IS NOW EXECUTING would handle this task.

Since %0 is the name of the batch file it is also possible that you may not have any further use for this piece of information. DOS assigns the entry after the batch filename to the %1 placeholder. Each subsequent piece of information is assigned to the next highest placeholder number, regardless of the order in which these placeholders are referenced within the batch file. For example, you might have batch files with the following entries:

```
TYPE %2.BAT
TYPE %1.BAT
```

If you name the file TRYORDER.BAT and execute it with this entry:

```
TRYORDER A B
```

DOS will type B.BAT then TYPE A.BAT since the first parameter A was assigned to %1 and the second parameter B was assigned to the placeholder %2.

■ ***NOTE:*** Although placeholders are assigned values based on their order in the command entry that starts the batch file and can be used in any sequence, it is best to use them in the same order within the batch file to prevent confusion.

Changing the Sequence of Execution

DOS normally executes the instructions in a batch file from the beginning. After executing one line, DOS moves to the next line and executes the commands there. This continues until DOS executes the last line in the batch file. You can use a GOTO command to change this normal execution flow. GOTO creates an unconditional branch. This means that control passes to another location regardless of conditions within the program.

GOTO is always followed by an indication of the place in the batch file that you want to branch to. This location is indicated as a label. DOS begins searches at the top of the batch file and looks for a line containing the same characters as the label preceded by a colon. Since the label with the colon preceding it must be on a line by itself DOS executes the command in the line following the label.

Skipping Around Code

You can use a GOTO instruction as a way to skip around certain code during testing. You can also combine the GOTO instruction with some of the logic

commands that allow you to check a condition and only branch to the new location in limited circumstances.

An example of a GOTO statement is included in the following batch file:

```
:START
ECHO OFF
BRANCH END
COPY *.* A:
DIR A:
:END
ECHO You have reached the end of the batch file
```

The location of the GOTO END instruction causes DOS to skip the COPY and DIR commands as shown in the diagram in Figure 13-1. The following points are important to understanding the batch file:

- START marks the beginning of the batch file.
- ECHO OFF stops subsequent commands from echoing to the screen.
- BRANCH END is an unconditional branch to END taken every time the batch file is executed.
- The COPY and DIR commands are never executed because the logic always branches to END where the final ECHO command is executed.

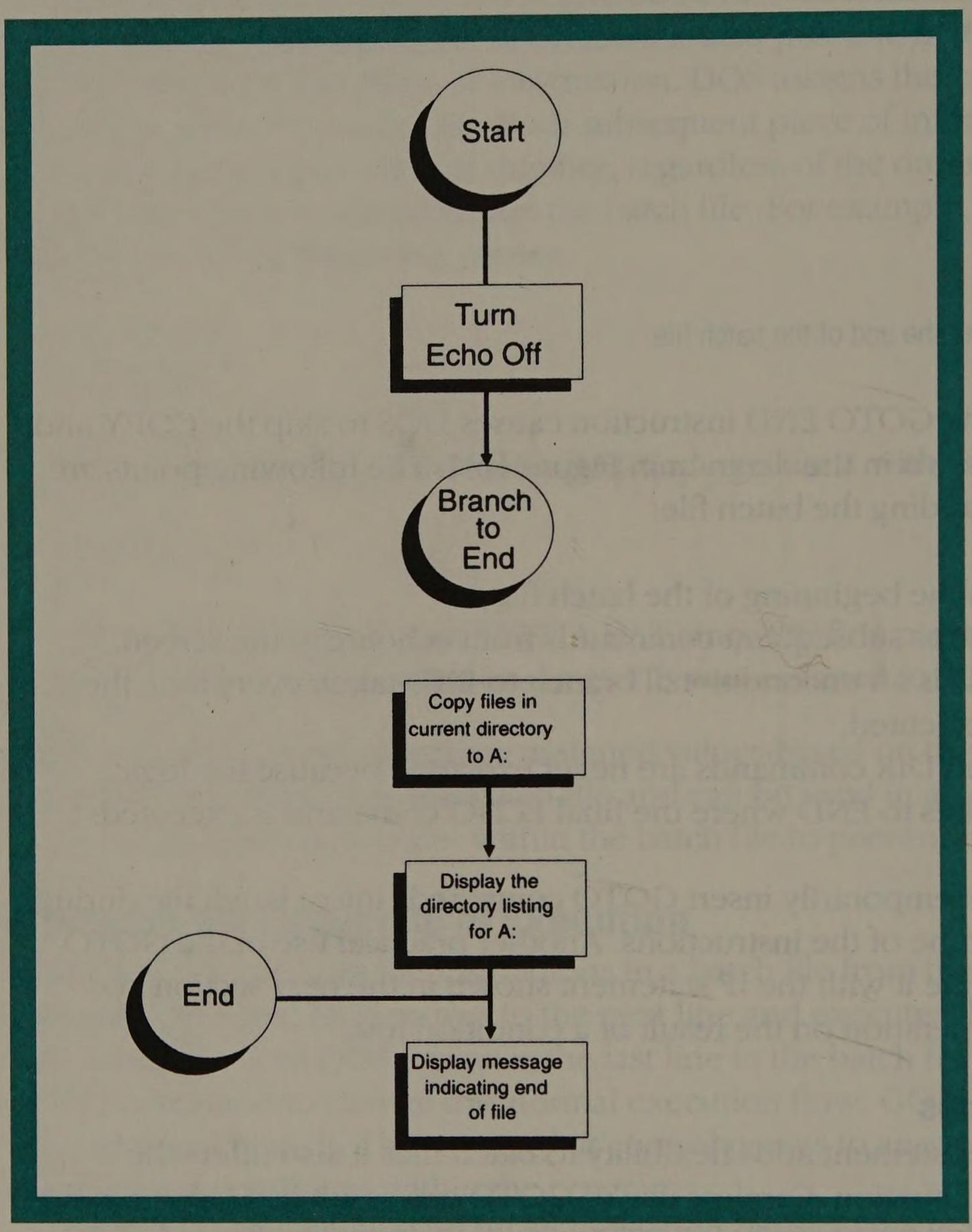
You may want to temporarily insert GOTO commands into a batch file during testing to skip over some of the instructions. Another practical use of the GOTO command is to combine it with the IF statement shown in the next section and base the branching operation on the result of a condition test.

Avoiding Endless Loops

Although the GOTO statement adds flexibility to batch files it also offers the potential for an error situation. Careless use of GOTO can result in a batch file that will execute forever. As an example, a GOTO statement placed at the end of a batch file that branches to the beginning of a file creates a loop which is endless. The following entries in a batch file would result in an endless loop between BEGIN and END:

```
:BEGIN
PRINT MASTER
TYPE MASTER
:END
GOTO BEGIN
```

Figure 13-1: Batch file with a GOTO that Skips Instructions



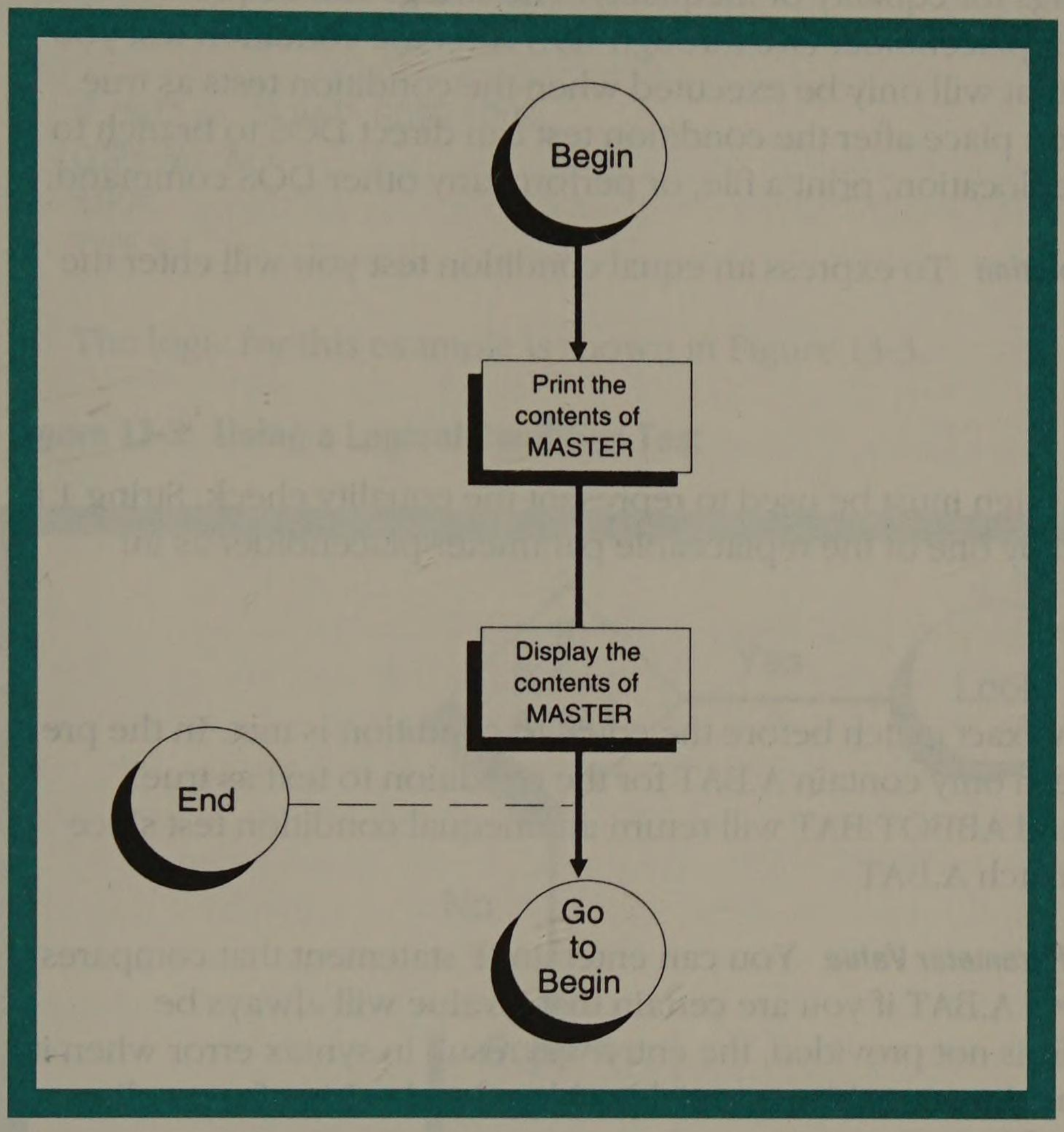
The location of the branch makes it possible to execute the COPY or DIR command.

The logic flow for this batch file looks like Figure 13-2.

After executing each instruction in the file (except the last) the final instruction GOTO BEGIN is executed. Each instruction is executed again and the cycle continues indefinitely.

Although you will want to avoid this situation, if it ever occurs accidentally, you can interrupt the file with CTRL-C. After interrupting the file you can edit it to remove or modify the GOTO statement causing the problem.

Figure 13-2: Endless Loop Within a Batch File



This batch file will execute forever.

Performing Condition Tests

The IF command allows you to look at current conditions and determine the desired action based on the result of the condition test. You can use IF to test for an equal condition, a not equal condition, the existence of a specific filename, and an error condition.

The parameter in the IF statement that follows the condition test is the command that will be executed if the condition that precedes it is true. You can combine IF with a GOTO instruction that creates a conditional branch which will only execute when the condition is true. You can also use other commands such as ECHO to displays a message if the condition is true or CALL which will again only function when the condition is true.

Testing Two Strings

You can test two strings for equality or inequality. The strings can be provided as quote enclosed text or placeholder (%0 through %9.) After the condition test you provide a command that will only be executed when the condition tests as true. The command that you place after the condition test can direct DOS to branch to instructions at another location, print a file, or perform any other DOS command.

Expressing an Equal Condition To express an equal condition test you will enter the IF statement as:

```
IF string1 == string2.
```

The double equal sign must be used to represent the equality check. String 1 is normally represented by one of the replaceable parameter placeholder as in:

```
IF %1 == A.BAT
```

There must be an exact match before the equal to condition is met. In the preceding example %1 can only contain A.BAT for the condition to text as true. Entries of *.BAT, A, and ABBOT.BAT will return an unequal condition test since they do not exactly match A.BAT.

Dealing with a Missing Parameter Value You can enter an IF statement that compares two strings as IF %1 == A.BAT if you are certain that a value will always be provided for %1. If %1 is not provided, the entry will result in syntax error when it is processed. You can eliminate this potential problem by checking for a null entry with an entry such as this:

```
IF "%1" == ""
```

The entry translates to the following when processed:

```
IF '' == ''
```

Thus, when %1 is empty an equal condition results. Without the quotes an empty %1 will cause a syntax error.

Another solution is to add a string of literal characters after %1. In this situation, quotes are not needed. You might enter:

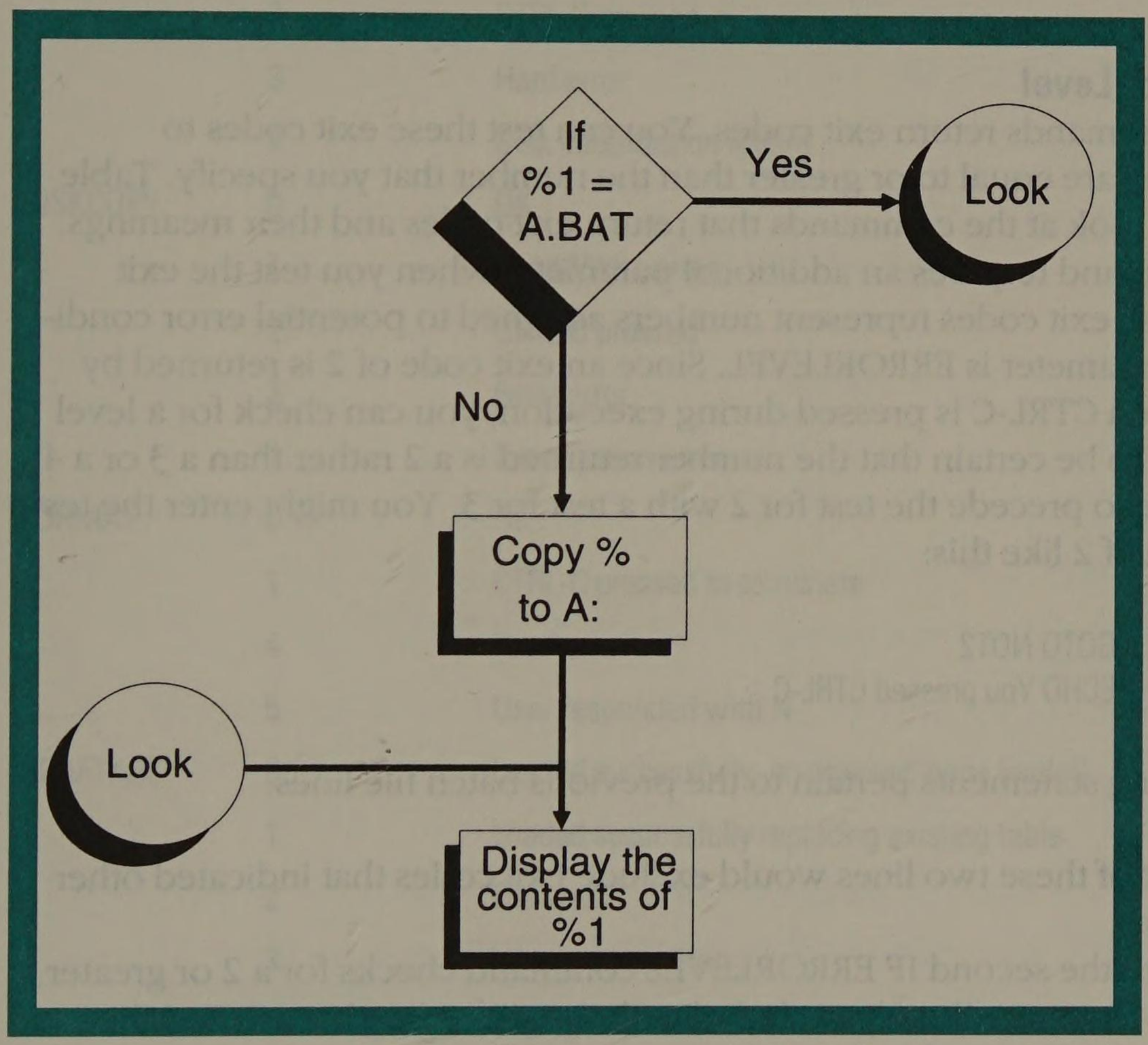
```
IF %1... == ...
```

Adding a Command The second part of the IF statement is a command. This parameter is always needed because it tells DOS the action to perform when the condition test is true. You might add a GOTO command after the IF as in:

```
IF "%1" == "A.BAT" GOTO LOOK
COPY %1 A:
:LOOK
TYPE %1
```

The logic for this example is shown in Figure 13-3.

Figure 13-3: Using a Logical Condition Test



The COPY command is only executed for files other than A.BAT.

The following statements apply to the example:

- LOOK must be a label on a section of the batch file code that DOS will branch to if the condition tests as true.

- In the batch file lines just shown, a true condition test causes DOS to skip the COPY command as it branches to LOOK.
- If %1 is equal to anything other than A.BAT the test will be false and the COPY will be completed before the TYPE command is executed.

Expressing an Unequal Condition You can negate an IF condition by adding the optional parameter NOT. If you wanted to branch to LOOK in situations where %1 was not equal to A.BAT you could write the condition test like this:

```
IF NOT %1 == A.BAT GOTO LOOK
```

The IF statement has exactly the opposite effect of the one shown in the previous section.

Testing the Error Level

Certain DOS commands return exit codes. You can test these exit codes to determine if they are equal to or greater than the number that you specify. Table 13-1 provides a look at the commands that return exit codes and their meanings.

The IF command requires an additional parameter when you test the exit code. Since these exit codes represent numbers assigned to potential error conditions the new parameter is ERRORLEVEL. Since an exit code of 2 is returned by DISKCOMP when CTRL-C is pressed during execution, you can check for a level of 2 or greater. To be certain that the number returned is a 2 rather than a 3 or a 4 you would have to precede the test for 2 with a test for 3. You might enter the test for an exit code of 2 like this:

```
IF ERRORLEVEL 3 GOTO NOT2
IF ERRORLEVEL 2 ECHO You pressed CTRL-C
```

The following statements pertain to the previous batch file lines:

- The first of these two lines would exclude exit codes that indicated other problems.
- Although the second IF ERRORLEVEL command checks for a 2 or greater, higher values are already excluded with the preceding branch and the message displays only when the exit code is a 2.

You can use NOT with the IF ERRORLEVEL command. The entry IF NOT ERRORLEVEL 2 GOTO OK, is true only if the exit code is not 2 or greater.

Table 13-1: DOS Exit Codes

Command	Code	Meaning
BACKUP	0	Backup successful
	1	No files to backup
	2	File sharing conflicts prevented some files from being backed up
	3	CTRL-C terminated
	4	Error terminated BACKUP
DISKCOMP	0	OK
	1	Did not compare as equal
	2	CTRL-C pressed
	3	Hard error
	4	Disk initialization failure
DISKCOPY	0	OK
	1	Read/Write error
	2	CTRL-C pressed
	3	Fatal error
	4	Initialization error
FORMAT	0	OK
	1	CTRL-C pressed to terminate
	4	Fatal error
	5	User responded with N
GRAFTABL	0	Loaded successfully; no previous page loaded
	1	Loaded successfully replacing existing table
	2	File error
	3	Wrong parameter
	4	Wrong version of DOS
KEYB	0	Loaded successfully
	1	Invalid syntax, keyboard code, or code page
	2	Keyboard definition file bad or replaced

Command	Code	Meaning
	4	Error occurred with CON device
	5	Requested code page not prepared
REPLACE	0	OK
	2	Source files not found
	3	Source file path not found
	5	Access to replace files denied
	8	Insufficient memory
	11	Wrong command syntax
	15	Invalid drive specified
RESTORE	0	OK
	1	No files for restore found
	3	CTRL-C terminated
	4	Error terminated command
SETVER	0	OK
	1	Invalid switch
	2	Invalid filename
	3	Insufficient memory
	4	Invalid version
	5	Specified entry not in table
	6	System files not found
	7	Invalid drive
	8	Too many parameters
	9	Missing parameters
	10	Error reading system files
	11	Version table corrupt
	12	System files do not support a version table
	13	Insufficient space in version table
	14	Error writing system files

Command	Code	Meaning
XCOPY	0	OK
	1	No files found
	2	CTRL-C pressed
	4	Initialization error, insufficient memory, invalid syntax, or invalid drive
	5	Disk write error

Testing for the Existence of a File

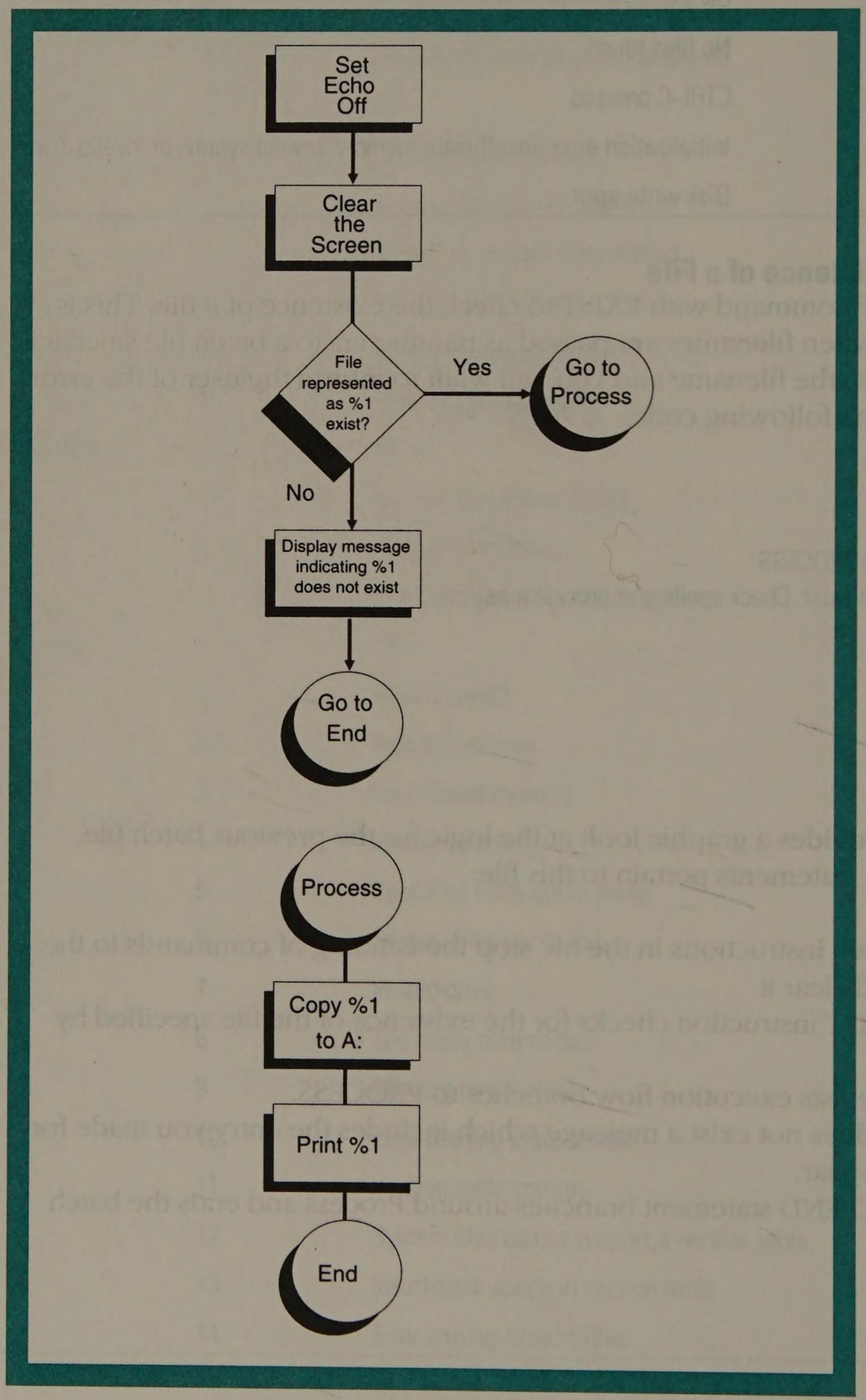
You can use the IF command with EXIST to check the existence of a file. This is especially useful when filenames are passed as parameters to a batch file since it is possible to misspell the filename and you will want to inform the user of the error. You might enter the following code:

```
ECHO OFF
CLS
IF EXIST %1 GOTO PROCESS
ECHO %1 does not exist. Check spelling or provide a path.
GOTO END
:PROCESS
COPY %1 A:
PRINT %1
:END
```

Figure 13-4 provides a graphic look at the logic for the previous batch file. The following statements pertain to this file:

- The first two instructions in the file stop the echoing of commands to the screen and clear it.
- The IF EXIST instruction checks for the existence of the file specified by %1.
- If the file exists execution flow branches to PROCESS.
- If the file does not exist a message which includes the entry you made for %1 will appear.
- The GOTO END statement branches around Process and ends the batch file.

Figure 13-4: Batch File that Checks for the Existence of a File with IF EXIST



The copy and print operations are only executed if the file exists.

You can check for the opposite condition with IF NOT EXIST. The logic is reversed and might look something like this:

```
ECHO OFF
CLS
IF NOT EXIST %1 GOTO MISSING
COPY %1 A:
PRINT %1
GOTO END
:MISSING
ECHO %1 does not exist. Check spelling or provide a path.
:END
```

Figure 13-5 provides a graphic illustration of the logic. The batch file executes the ECHO command at MISSING when the file specified with the parameter for the %1 placeholder does not exist; it then ends. If the NOT EXIST condition is not true the COPY and PRINT commands are executed before the batch file branches around the ECHO command at MISSING.

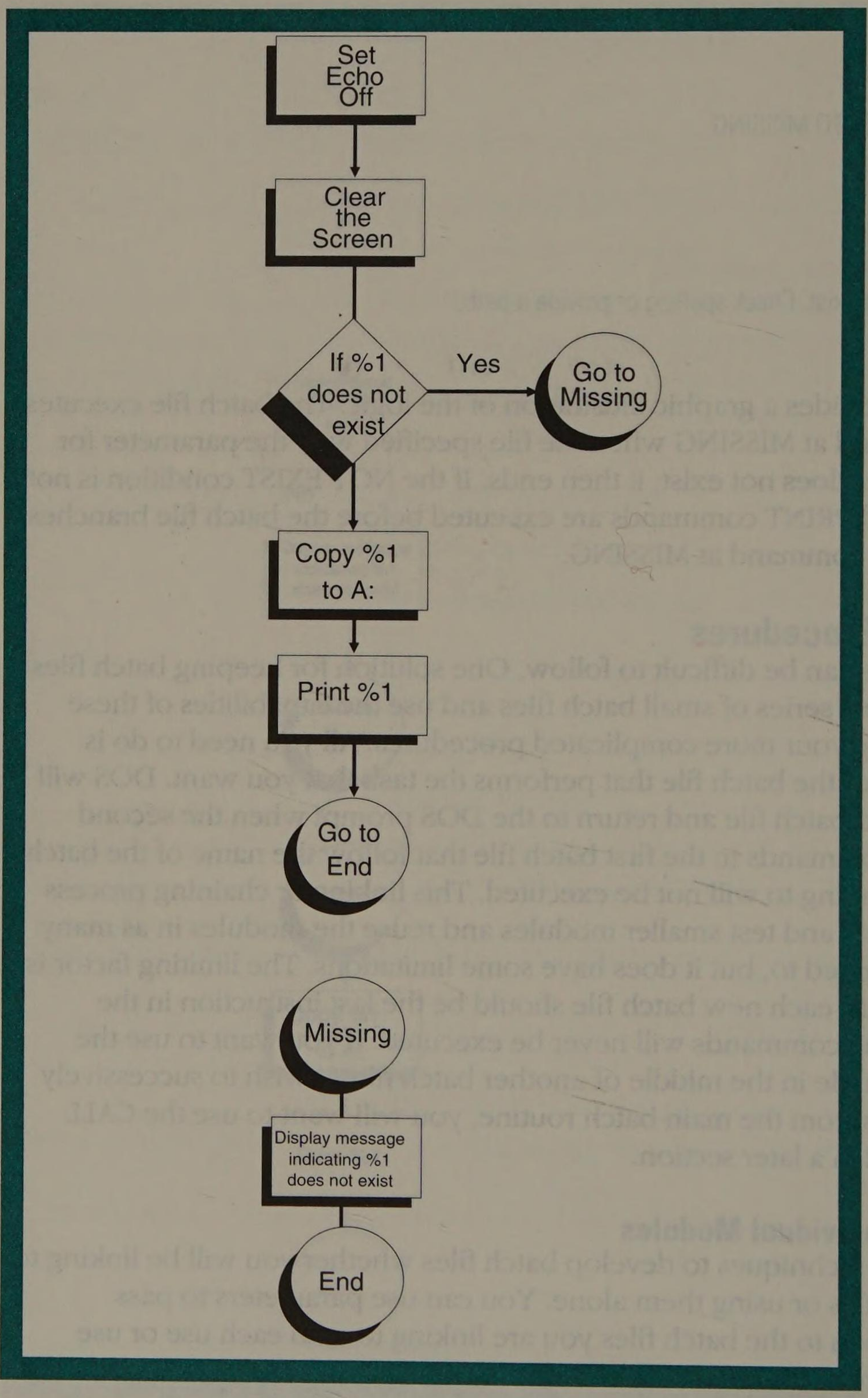
Linking Batch Procedures

Lengthy batch files can be difficult to follow. One solution for keeping batch files short is to develop a series of small batch files and use the capabilities of these files when building your more complicated procedures. All you need to do is include the name of the batch file that performs the task that you want. DOS will execute the named batch file and return to the DOS prompt when the second batch file ends. Commands in the first batch file that follow the name of the batch file that you are linking to will not be executed. This linking or chaining process allows you to create and test smaller modules and reuse the modules in as many batch files as you need to, but it does have some limitations. The limiting factor is that the reference to each new batch file should be the last instruction in the current file or some commands will never be executed. If you want to use the features of a batch file in the middle of another batch file or wish to successively make several links from the main batch routine, you will want to use the CALL features discussed in a later section.

Developing the Individual Modules

You use the same techniques to develop batch files whether you will be linking to them from other files or using them alone. You can use parameters to pass different information to the batch files you are linking to with each use or use them without change each time.

Figure 13-5: Another Way to Use IF EXIST



Processing is skipped if the file does not exist and a message is displayed.

One good reason for using a separate batch file for part of what you want to do is that you already have the batch file. Rather than copying its instructions into another batch file and reworking the logic, you can simply link to it if it is the first task you want to accomplish.

You might already have a batch file to create a backup of all the .BAK files that WordStar creates for backup. Rather than adding instructions for deleting these files once they have been backed up you might want to create a reference to a batch file that already performs this task. The batch file for backing up the files might be named BKUPBAK and look like this:

```
ECHO OFF
CLS
ECHO WordStar files with a .BAK extensions will be written to a backup disk. Place a blank disk in drive A
PAUSE
BACKUP C:\WS\*.BAK A:
```

The batch file that deletes all the .BAK files from the current directory is named ELIMBAK and contains these instructions:

```
ECHO All WordStar .BAK files in the current directory will be deleted
ECHO Press CTRL-C to stop or any other key to continue
DEL *.BAK
```

Combining the Pieces

Rather than add the instructions to delete the .BAK files to the BKUP batch file you can execute ELIMBAK from within the batch file. Using this simple link procedure, the flow of execution will change to ELIMBAK and not return to the original batch file, making it important that there not be any commands in the batch file you need to execute after eliminating the files.

```
ECHO OFF
CLS
ECHO WordStar files with a .BAK extensions will be written to a backup disk. Place a blank disk in drive A
PAUSE
BACKUP C:\WS\*.BAK A:
REM Check for unsuccessful conclusion
IF ERROR LEVEL 2 GOTO PROBLEM
REM Successful completion or no files found that match backup request
ELIMBAK
REM Only executed when backup is not successful
```

```
:PROBLEM
ECHO Backup was not successful - no files were deleted
```

Figure 13-6 provides a logic map for the new batch file. You will want to note the following features of the commands in the file:

- The first two commands stop the echo process and clear the screen to allow later messages to display at the top of the screen.
- The third command displays two messages to the user of the batch file.
- The PAUSE command waits until the user presses a key before continuing.
- The BACKUP procedure is completed.
- The REM statement explains that the IF ERRORLEVEL statement is checking for a successful conclusion.
- The IF ERRORLEVEL statement looks for exit codes of 2 or greater, branching to PROBLEM to indicate a problem with the backup.
- With a successful completion, ELIMBAK is executed next. Further instructions will not be executed from this file since the processing of batch file instructions will end in ELIMBAK. The specific instructions in ELIMBAK were shown previously.

Using CALL to Execute Other Batch Files

In the preceding section you looked at linking to the instructions in a second batch file as you finished processing the first batch file. Since control did not return to the first batch file it was important to be finished with the commands in the first file before moving on to the second file. A second method of executing the instructions in another batch file is a CALL command. With CALL you can execute the commands in another batch file but when this second batch file completes, control is returned to the first batch file at the instruction following the CALL statement and the first batch file resumes execution.

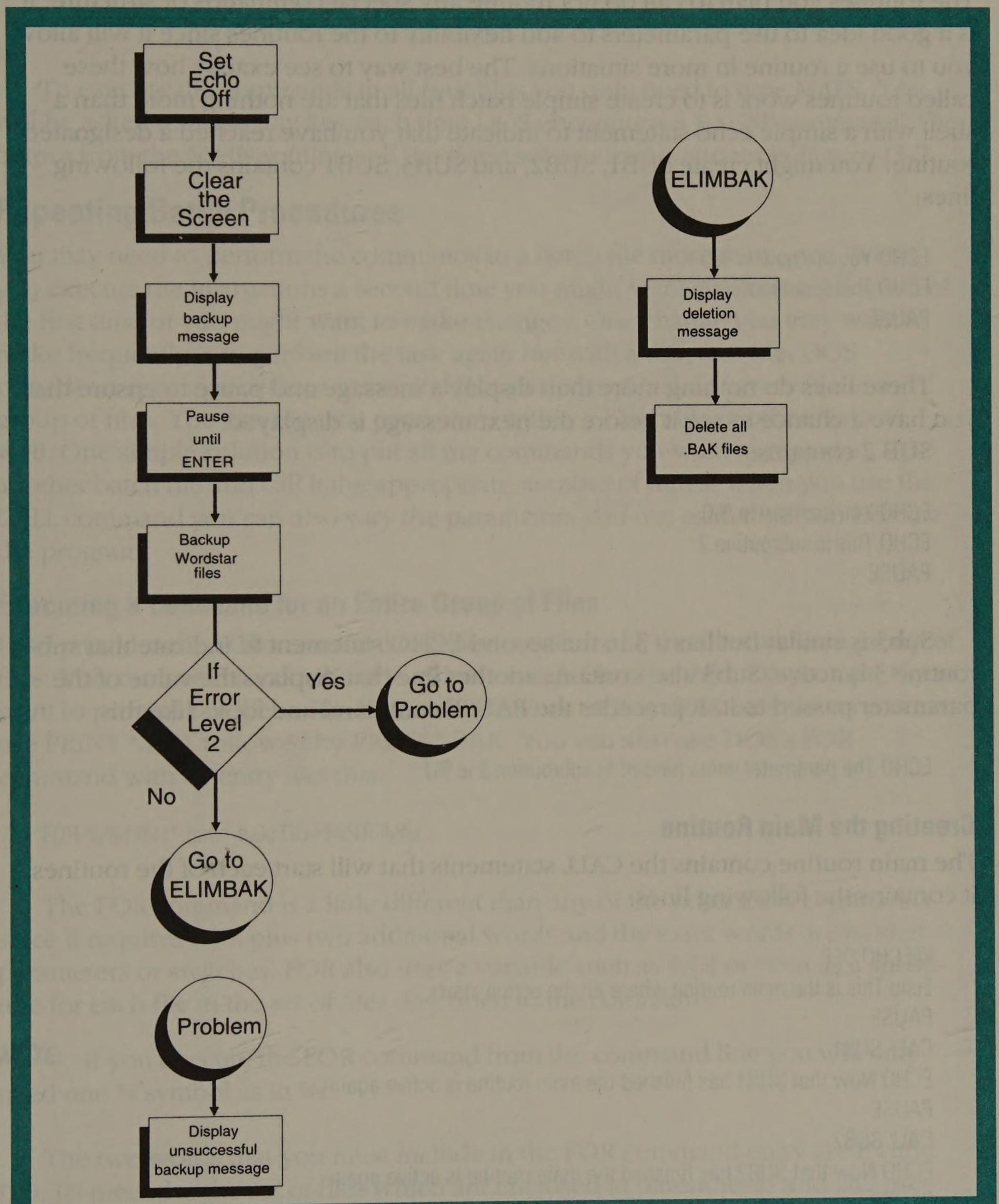
You can enter a call statement that starts a batch file named PRNT like this:

```
CALL PRNT
```

If the PRNT command uses parameters, you can add them. If PRNT expects a parameter presenting a filename, the call statement might look like this:

```
CALL PRNT ABC.BAT
```

Figure 13-6: Linking Two Batch Files



Creating the Routines that Will be Called

The routines you plan to call do not require any special commands or structure. It is a good idea to use parameters to add flexibility to the routines since it will allow you to use a routine in more situations. The best way to see exactly how these called routines work is to create simple batch files that are nothing more than a shell with a simple echo statement to indicate that you have reached a designated routine. You might create SUB1, SUB2, and SUB3. SUB1 contains the following lines:

```
ECHO You are now in %0
ECHO This is subroutine 1
PAUSE
```

These lines do nothing more than display a message and pause to ensure that you have a chance to see it before the next message is displayed.

SUB 2 contains:

```
ECHO You are now in %0
ECHO This is subroutine 2
PAUSE
```

Sub3 is similar but has a 3 in the second ECHO statement to indicate that subroutine 3 is active. Sub3 also contains another line that displays the value of the parameter passed to it. It precedes the PAUSE command and looks like this:

```
ECHO The parameter value passed to subroutine 3 is %1
```

Creating the Main Routine

The main routine contains the CALL statements that will start each of the routines. It contains the following lines:

```
@ECHO OFF
Echo This is the main routine where all the action starts
PAUSE
CALL SUB1
ECHO Now that SUB1 has finished the main routine is active again
PAUSE
CALL SUB2
ECHO Now that SUB2 has finished the main routine is active again
PAUSE
```

```
CALL SUB3 TRYTHIS
ECHO Now that SUB3 has finished the main routine is active again
ECHO Since there are no more commands in the main routine it will also end
```

To execute the commands in all four files you only need to type MAIN. You will be asked to press any key each time DOS encounters a PAUSE command. The output from the MAIN routine and the three subroutines is shown in Figure 13-7.

Repeating Batch Procedures

You may need to perform the commands in a batch file more than once. When you execute the instructions a second time you might want an exact replication of the first time or you might want to make changes. One change you may want to make frequently is to perform the task again but with a different file. DOS provides a special repetitive command that performs a task for every file in a group of files. You can repeat a procedure within a batch file in other ways as well. One simple solution is to put all the commands you want to repeat in another batch file and call it the appropriate number of times. When you use the CALL command you can also vary the parameters you use each time you execute the program.

Executing a Command for an Entire Group of Files

DOS provides the FOR command to repeat a task for all the files within a group of files. You can use this command in either a batch file or at the DOS prompt. If you want to print all the files with .BAT extensions as well as .BAK extensions you can use PRINT *.BAT followed by PRINT *.BAK. You can also use DOS's FOR command with an entry like this:

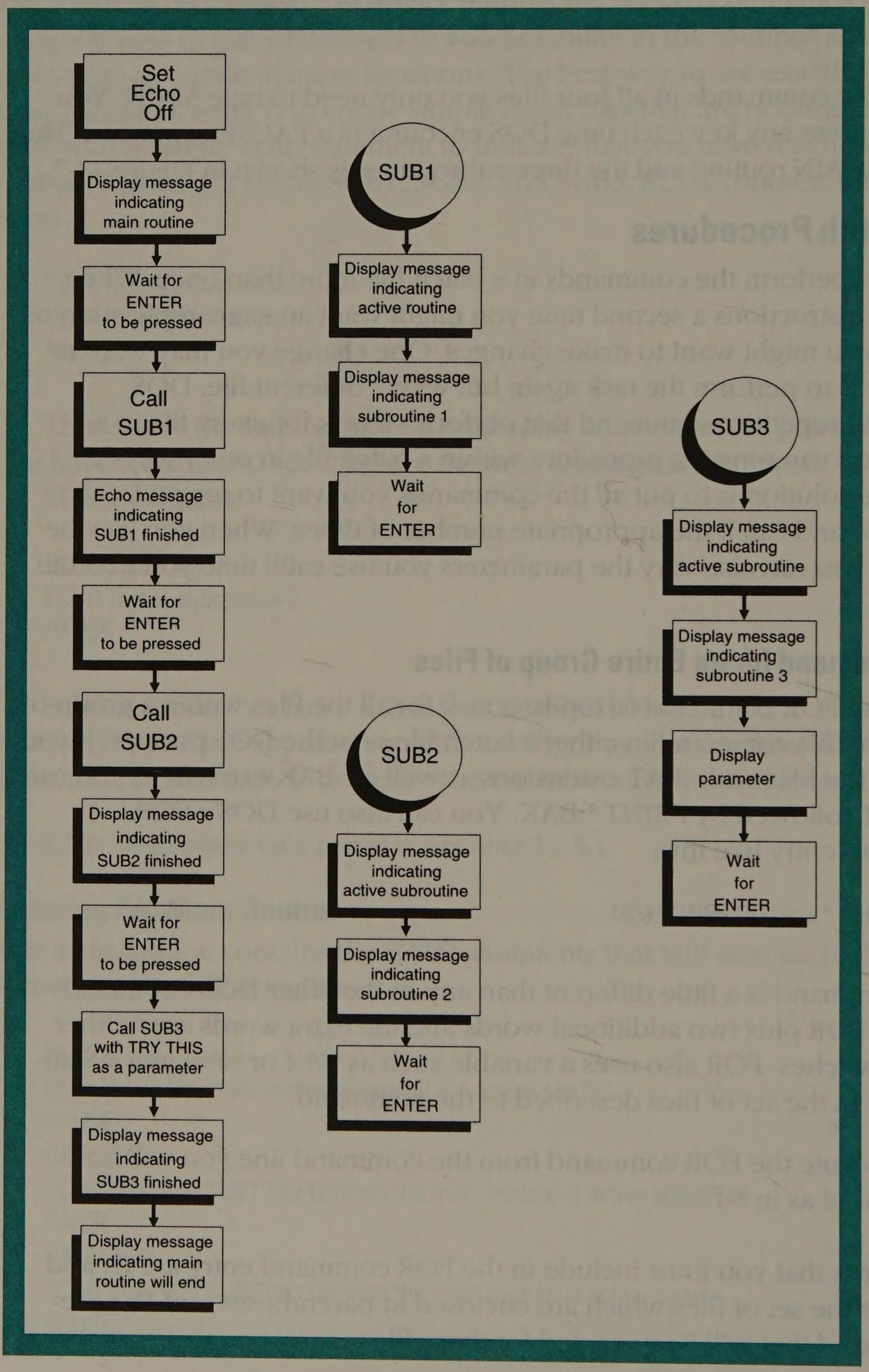
```
FOR %%f IN (*.bat,*.bak) DO PRINT %%f
```

The FOR command is a little different than any of the other DOS commands since it requires FOR plus two additional words and the extra words are neither parameters or switches. FOR also uses a variable such as %%f or %%a as a substitute for each file in the set of files described to the command.

■ ***NOTE:*** If you execute the FOR command from the command line you will only need one % symbol as in %f.

The two words that you must include in the FOR command entry are IN and DO. IN precedes the set of files which are enclosed in parentheses and DO precedes the command that will be executed for these files.

Figure 13-7: Messages Displayed by the MAIN Routine and SUB1, SUB2, and SUB3


Set Echo Off
Display message indicating main routine
Wait for ENTER to be pressed
Call SUB1
Echo message indicating SUB1 finished
Wait for ENTER to be pressed
Call SUB2
Display message indicating SUB2 finished
Wait for ENTER to be pressed
Call SUB3 with TRY THIS as a parameter
Display message indicating SUB3 finished
Display message indicating main routine will end
SUB1
Display message indicating active routine
Display message indicating subroutine 1
Wait for ENTER
SUB2
Display message indicating active subroutine
Display message indicating subroutine 2
Wait for ENTER
SUB3
Display message indicating active subroutine
Display message indicating subroutine 3
Display parameter
Wait for ENTER

All three subroutines are executed automatically when MAIN is executed.

Specifying the Set of Files The FOR command accepts a set of files. The action specified at the end of the FOR statement will be executed for each file in the set. In the sample FOR entry, (*.bat,*.bak) represent a set of all .BAT and .BAK files from the current directory. The command at the end of the FOR statement will execute until all of the entries in the set are processed.

```
REM  This file prints all the batch files in the current
REM  directory. For your output to be useful, you should include
REM  REM statements in every batch file.
FOR %%f IN (*.bat,*.bak) DO PRINT %%f
```

Figure 13-8 shows the flow of execution as DOS executes this batch file. Although there is only one instruction that does any work, the substitution of each value within the set makes this a powerful command that can accomplish a substantial amount of work for you. The FOR command is equivalent to the following instructions with the values provided in the set:

```
PRINT *.BAT
PRINT *.BAK
```

Using SET

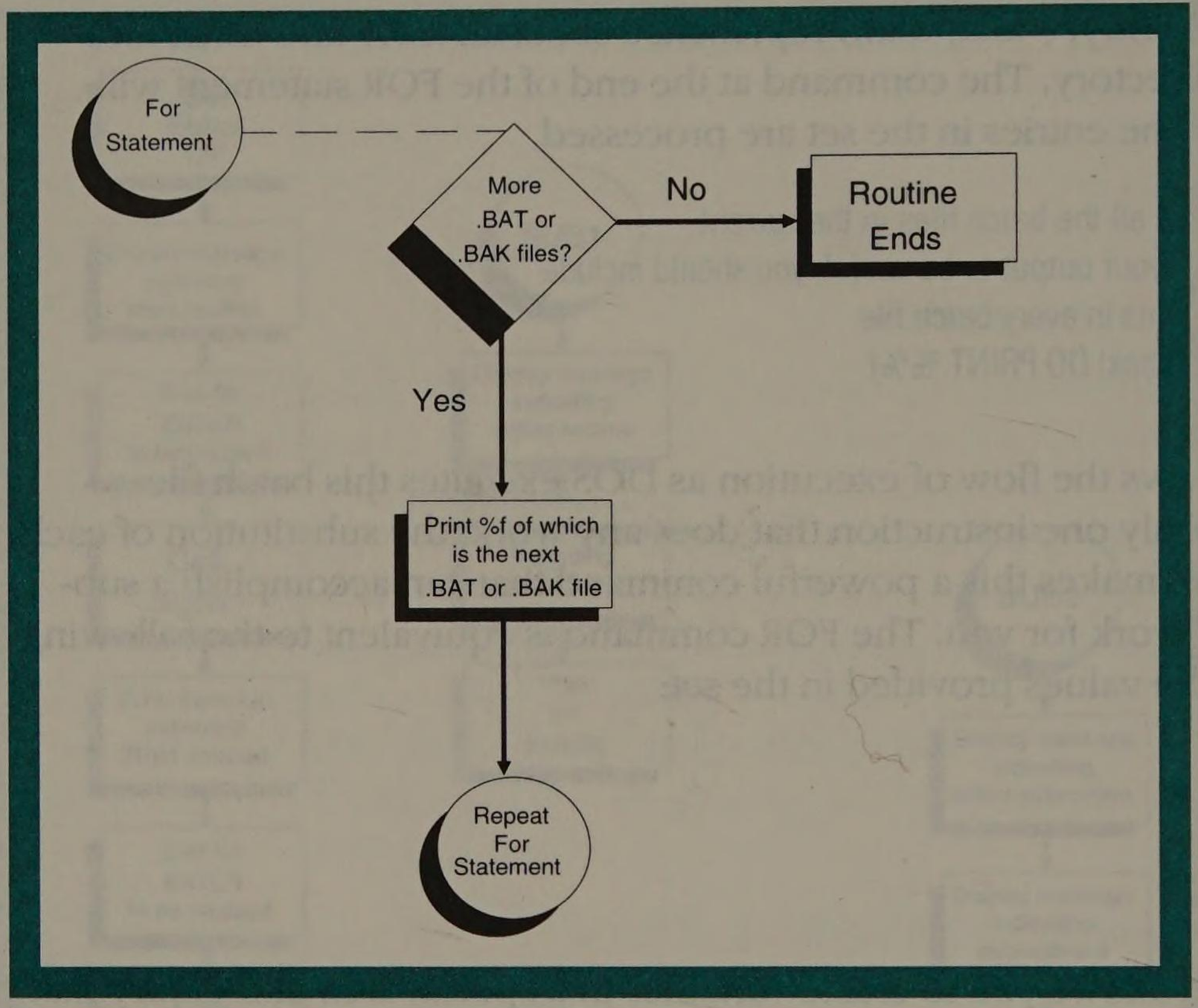
The SET command allows you to create variables in a special area of memory that DOS reserves to keep track of information relating to the environment in which DOS is running. DOS stores this information in *environmental variables.* Two of these variables are the prompt and path, but you can create additional variables of your own. You can store information in these variables and check the contents of the variables in order to direct the execution of your program. You can also look at the current settings of all environmental variables that are defined.

Displaying the Current Environmental Variables

If you type **SET** at the DOS prompt and press ENTER, DOS will display the current environmental variables and their present settings. A typical display might look something like this:

```
COMSPEC=C:\COMMAND.COM
PATH=C:\DOS;C:\WS5;C:\WP51\
APPEND=C:\DOS
PROMPT=Enter 'EXIT' to return to WordPerfect $_$P$G
```

Figure 13-8: Execution Flow for a FOR Command



One FOR command can perform many tasks.

Creating Variables Yourself

You can add variables of your own to the memory space that DOS reserves for environmental variables. You might create variables with the name of the application and current file with these instructions:

```
SET APP=WORDPERFECT
SET FILE=C:\WP51\DOS\DOSCH13
```

These two commands place these entries in the environmental variable space:

```
APP=WORDPERFECT
FILE=C:\WP51\DOS\DOSCH13
```

You can test the strings in these variables with an IF statement and take appropriate action based on the result of the condition test. If you use an environmental variable in an IF command in a batch file, you must enclose it between two % symbols as in **IF %APP%==WORDPERFECT GOTO WORD**.

Removing an Environmental Variable

The SET command can be used to remove variables as well as to add them. Entering the name of the variable and an equal sign (=) without a new value removes the variable. If you enter **SET APP=** and press ENTER, APP will no longer be defined as an environmental variable.

Shifting Parameters

You can repeat the commands in a batch file for a number of different parameters without duplicating the commands with different placeholders. The SHIFT command allows you to shift the parameter values you provide when you execute a batch file into a lower numbered placeholder. You might enter a batch file request with three parameters like this:

```
HELLO John Paul Harry
```

Hello would be assigned to %0, John to %1, Paul to %2, and Harry to %3. You could write a batch file to say hello to each of the three individuals like this:

```
ECHO OFF
CLS
ECHO Hello, %1!
ECHO Hello, %2!
ECHO Hello, %3!
```

The output will displays as:

```
Hello, John!
Hello, Paul!
Hello, Harry!
```

Another approach that produces the same output shifts the value for %2 and %3 into %1 after the task is performed for %1. The batch file which produces the same result with shift would be executed in the same way but the code might look like this:

```
ECHO OFF
CLS
:TOP
ECHO Hello, %1!
SHIFT
IF NOT "%1" == "" GOTO TOP
```

When the code is entered the first time, Hello, John! displays. The SHIFT command moves Paul to %1 making Hello, Paul display after the GOTO command causes the branch to TOP and the second execution to the ECHO command. When SHIFT is executed again, Harry is shifted into %1.

The SHIFT command allows you to extend the usefulness of replaceable parameters. Each time you use SHIFT, %1 is shifted to %0, %2 to %1, %9 to %8, and entries beyond %9 are also shifted one place to the left. This means that you can enter values that would be assigned to placeholders higher than %9. Even though you cannot add higher placeholders to the batch file, they will not be needed since the SHIFT command can shift the higher entries to %9 or a lower number.

At a Glance

In this chapter you learned to use many new batch file commands that provide programming capabilities for your batch files. The following commands were covered:

- **Test two strings for equality—IF**
- **Change the execution sequence of instructions—GOTO**
- **Execute a batch procedure and return—CALL**
- **Use variables in a batch file—%0 through %9**
- **Shift variable values to expand beyond the limit of 10—SHIFT**
- **Repeat a procedure—FOR**

Chapter 14

Controlling Devices and Output

To make it easier to accomplish DOS tasks, most commands are entered through the default input device, the keyboard. Depending on the command, output is automatically routed to the screen or the name of the file that you specify. DOS allows you to override its defaults and receive or send input or output from other devices. DOS uses standard device names for accessing these other alternatives. Every time you enter a command at the DOS prompt, DOS checks to see if command parameters are device names before attempting to use them as filenames.

DOS provides filters that allow additional control over your output. You can use a SORT filter for resequencing data stored in a text file. You can control the display of output on the screen with the MORE filter command. MORE restricts the data displayed at one time to the amount that will fit on one screen. You can also locate a specific piece of information that you want with the FIND filter. In addition to learning the filters themselves, you will also learn to use *pipes* to send the output from one filter to another filter command. As an example, you might want to select matching records and then sort the records that match your text before displaying them. You can do this by piping from the FIND filter to the SORT filter.

What You'll Learn

After reading this chapter, you will be able to:

- **Describe the device for each device name**
- **Send output from commands to a device other than the default device**
- **Receive input from a device other than the keyboard**
- **Sort text file data**
- **Limit the display of information to one screen at a time**
- **Find matching text in text files**

Understanding Device Names

DOS uses default devices for input and output. Default devices represent the standard locations where DOS expects to either get or put information. For example, DOS expects input for commands to come from the keyboard unless you override this default with another entry. When input data is specified as a parameter for a command, it is usually a file, although other devices, such as the keyboard, can also be specified.

There are two kinds of output from commands: informational messages and prompts, and, for most commands, the results of the command itself. Although the screen is used for output messages, there are other possibilities for results. Some commands display their output results on the screen and others on the printer. For some commands, you normally specify file names for both input and output. For these commands you can use device names for input and output locations. Even when the command does not specify an input or output location as a parameter, you can use redirection symbols to indicate the location of this information in a file or other device. Specifying the device name in a command works any time the command expects you to provide an input or output location.

Using Valid Device Specifications

DOS supports the use of four different device specifications. These devices allow you to use your keyboard, screen, serial and parallel ports, as well as a dummy location that essentially throws away the command output.

You can indicate the keyboard and screen with CON. CON is an abbreviation for console. This device specification is unique, since it can represent input from the keyboard or output to the screen. If input is expected in the command where you enter CON, DOS will expect you to type the input at the keyboard. If the command expects an output location, the screen will be used for display if you specify CON.

The serial or communications port on your computer is indicated as COM1 or AUX. If you want to direct output to the device or terminal attached to this port you will use AUX or COM1 in the command. DOS 5.0 supports as many as three additional serial ports with entries of COM2, COM3, and COM4.

To direct output to a parallel port, use LPT1 or PRN for the first port. Since this is the location where most systems have a printer attached, PRN is an appropriate selection for this device name. Additional parallel ports might be LPT2 and LPT3.

If you want to throw away the output from a command, you can use the device NUL. You might use NUL when you want to see the message DOS displays when output is successful, as in COPY TEXT NUL. (DOS will indicate that 1 file has been copied even though a file is not created.) NUL is especially useful in

batch files. You might consider using NUL as the print device to speed up the testing of logic in a batch file without taking the time to print.

DOS scans your entries on the command line to see if it recognizes one of its special device names, and uses them rather than creating or searching for a disk file with the same name.

Table 14-1 summarizes these device names and the devices they represent for you. These are the same devices that you will use for the redirection options discussed later. First, you will want to take a look at a few of these device names used in commands.

Table 14-1: Device Names

Device	Represents
CON	Console—Your computer's keyboard and display screen
AUX or COM1, COM2, COM3, COM4	Serial ports for connecting a serial printer or communications device such as a modem
PRN or LPT1, LPT2, LPT3	Parallel ports for connecting a device such as a parallel printer
NUL	DOS discards information sent to this device

Using the Console as a Device Command input and output messages use your computer keyboard and screen as their standard location. Taken together, the keyboard and screen are referred to as CON. In Chapter 12 you used CON in place of a source file in the COPY command. This allowed you to type input at your keyboard for storage in the target file. If you wanted to copy from the keyboard to a file named BUDGET.TXT you would type **COPY CON BUDGET.TXT**. DOS would then store each line that you typed in the file BUDGET.TXT. To finally end the entry of data, you would press CTRL-Z or F6 and follow that with ENTER. The "1 file copied" message would then display to indicate that the copy operation was complete.

Printing a File with the COPY Command

When you use the PRINT command to print a file, the print output is routed to the spooler, and the spooler sends the information to the printer. If you need to bypass the spooler, you can use PRN in place of a target file in your copy statement. Your entry might look like this if you wanted to print BUDGET.TXT:

```
COPY BUDGET.TXT PRN
```

You can use this approach to print a file containing printer codes, including one that contains graphics. Where PRINT will stop at the first representation that looks like an end of file, COPY will continue to transmit the number of characters specified by the file size. This is especially helpful when you are printing files containing graphics, since the graphics file may contain numerous entries that look like an end of file indicator. You will also want to use the COPY command instead of PRINT when you do not want to use RAM for the RAM-resident PRINT program. Part of the PRINT command remains in RAM after you use it, while COPY does not. This is especially important if you are using the DOS shell in a program such as WordPerfect or 1-2-3, since when you return to these programs by typing **EXIT** and pressing ENTER, the program needs the RAM that PRINT would otherwise take up.

Redirecting Input and Output

Unless you change the standard input and output device with CTTY, DOS accepts input from the keyboard and displays output to your screen. As mentioned earlier, this makes CON the default or standard device for the commands you enter and the messages and prompts that DOS displays for you. You can use redirection symbols to redirect this input and output to or from another file or a device such as your printer.

■ ***NOTE:*** Be careful about redirecting command output when the command will prompt you for information. For example, if you redirect the output of the UNDELETE command, you will not see the information DOS displays about the files you can potentially undelete and the prompt where DOS is asking you if you want each file undeleted.

Placing the Output in a File

If you want to retain a copy of the output from a command, you might want to write the information to a file rather than your screen. To redirect your output, place a greater than symbol (>) after the command and follow it with the file or device name to which the output should be written. For example, to write the output from a directory listing of drive A to a disk file named DRIVEA, you type:

```
DIR A: >DRIVEA
```

The file DRIVEA will be created on the current drive, if it does not already exist, and the directory listing that normally displays on your screen will be written

to this file. If DRIVEA already exists, its current contents will be replaced with this directory listing.

- ***NOTE:*** Think of the redirection symbol (>) as an arrow head that points in the direction that output will be sent. You can think of the previous command as looking like this:

```
DIR A: >DRIVEA
```

If you want to redirect the directory listing to your printer you can type:

```
DIR A: >PRN
```

You can use this feature for any of the DOS commands. TREE and CHKDSK are two additional options shown here:

```
TREE A: >TREELIST
CHKDSK B: >CHKOUT
```

Notice in these last two entries that there is no space before the redirection symbol. The command works the same whether or not you use the space.

Adding the Output to the End of a File

If you have a series of tasks to perform and want all the output written to one file, you will need to use the redirection option that appends data. A double greater than symbol (>>) is the appropriate selection if you want to add to the end of existing data. If you plan to list the directories from two different drives in one file, you can use the regular redirection option for the first drive then use the APPEND option for the other drive. Your entries might look something like this if you are copying the directory information to a file named ALLINFO:

```
DIR A: > ALLINFO
DIR B: >> ALLINFO
```

The first command writes the data to a new file or replaces the contents of the existing file, ALLINFO. The second command adds data to that end of the file. Sample directory listings for A: and B: are shown in Figures 14-1 and 14-2. Figure 14-3 shows what the display would look like if you entered **TYPE ALLINFO** after completing both directory commands on these sample listings.

Figure 14-1: Directory Listing for Drive A

```
C:\HSG>DIR A:

 Volume in drive A has no label
 Directory of A:\

DOSAPPB          42011 09-15-90  12:23p
DOSAPPF          78065 09-19-90   1:04p
DOSCMD   BAT        27 09-24-90  12:47p
MOVEDATA BAT        12 09-20-90  12:33p
DOSCH12          33659 09-24-90   1:35p
       5 file(s)     153774 bytes
                    1302016 bytes free

C:\HSG>
```

These five files are stored on drive A.

Figure 14-2: Directory Listing for Drive B

```
C:\HSG>DIR B:

 Volume in drive B has no label
 Directory of B:\

WFBG     SYN    325330 01-15-87   8:12a
MIRROR   FIL      6144 10-05-90   4:36p
PCTRACKR DEL      4554 10-05-90   4:37p
       3 file(s)     336028 bytes
                      24576 bytes free

C:\HSG>
```

These are three files on drive B.

■ ***NOTE:*** When you direct command output to a file, the file contains text that you can edit with your word processing package. You can use the word processor to change text features or fonts and to remove lines that are not of any interest.

Obtaining Input from Another Device

Just as you can send output to a device other than the default device, you can redirect input to come from a different source. The less than symbol (<) is used to represent input for a command. If you add an imaginary line after the less than symbol you will have a mental picture of an arrow pointing toward the command.

Since the filename comes next, you can think of this arrow as directing the file toward the command.

The filter commands FIND and SORT are two commands where redirected input are often specified. To tell DOS to sort the data in the file TEXT, you would enter SORT <TEXT. After sorting the lines in TEXT, DOS displays them on your screen, since you have not directed the output to another location.

Figure 14-3: Contents of ALLINFO

```
C:\HSG>TYPE ALLINFO

 Volume in drive A has no label
 Directory of A:\

DOSAPPB          42011 09-15-90  12:23p
DOSAPPF          78065 09-19-90   1:04p
DOSCMD   BAT        27 09-24-90  12:47p
MOVEDATA BAT        12 09-20-90  12:33p
DOSCH12          33659 09-24-90   1:35p
        5 file(s)     153774 bytes
                     1302016 bytes free

 Volume in drive B has no label
 Directory of B:\

WFBG     SYN    325330 01-15-87   8:12a
MIRROR   FIL      6144 10-05-90   4:36p
PCTRACKR DEL      4554 10-05-90   4:37p
        3 file(s)     336028 bytes
                       24576 bytes free

C:\HSG>
```

The file ALLINFO contains the directory listing for both drives.

Using CTTY to Change the Default Input/Output Device

The CTTY command can change the device that DOS uses for the default input and output device. For example, if you want to control a computer though a remote terminal device which is connected through the first serial port (COM1 or AUX), you would enter:

```
CTTY AUX
```

After you execute this command, DOS will accept input that would normally be from the keyboard from the auxiliary device, and send output normally sent to the screen to the auxiliary device. When you want to return to using the keyboard and screen for input and output, enter:

```
CTTY CON
```

Before you use the CTTY command, you must make any modifications to the serial port you want to be in effect after you make the change. This is done with the MODE command, which is described in Chapter 9.

Using Filters

Filters are used to select information or change its presentation. You can use the FIND command to extract lines from a file that match text you enter. The MORE command is a filter that slows the presentation of information to one screen at a time. The SORT command is also a filter. SORT rearranges the lines in a file according to the entries in a specific column of the file.

Locating Text

The FIND command looks for a series of characters in each line of a text file. For each line containing the text that you are searching for, DOS displays the line on your screen unless it is directed to another file or device.

The following lines of text are stored in the file INVENTRY:

```
Table        4    ABC Supply
Chair        2    Office Masters
Credenza     1    ABC Supply
Bookcase     9    Woodcare, Inc.
```

To locate all the records where the supplier is ABC Supply, you would enter:

```
FIND "ABC Supply" INVENTRY
```

The text you are searching for must be enclosed in single or double quotes. Unless you use the special /i switch discussed later, the search is case-sensitive, distinguishing between upper- and lowercase characters. The matching records are displayed on the screen like this:

```
---------- INVENTRY
Table        4    ABC Supply
Credenza     1    ABC Supply
```

The redirection symbol (>) allows you to route these matching lines to a printer or another file, as in FIND "ABC" INVENTRY >JUSTABC or FIND "ABC" INVENTRY >LPT1. The first example routes the matching records to the file JUSTABC and the second prints the matching records.

Exploring Special FIND Options

You can extend FIND beyond the basic search capability. For example, you can establish a search that is not case-sensitive, or one that negates the search condition. You can also process a number of files at once, count the matching lines, and display line numbers for the matching lines.

Looking In Multiple Files Although FIND does not support the use of the * and ? wildcards, you can look in multiple files by providing more than one filename to FIND. To look in multiple files all you need to do is place all the file names in the command after the quote-enclosed string you are searching for. If you have one file for each week in September, for instance, you can search all the files for lines that contain "Jackson". Assuming the file names are WEEK9_1, WEEK9_8, WEEK9_15, WEEK9_22, and WEEK9_29, you can search all the files with this entry:

```
FIND "Jackson" WEEK9_1 WEEK9_8 WEEK9_15 WEEK9_22 Week9_29
```

Given that each of the files contains at least one matching record, your output might look something like that in Figure 14-4. DOS always displays the file of the matching record above the group of lines displayed from the file.

Counting Matching Lines You can use the /c switch with find to obtain a count of the lines that match your entry. Rather than displaying each matching line, a number is displayed. After a FIND command that found two matches in a file named INVENTRY, the display looks like this:

```
---------- INVENTRY: 2
```

Making the Search Not Case Sensitive Normally, the FIND command distinguishes between letters that are in upper and in lower case. You can use the FIND command to find text that is both by using the /i switch. As an example, if you want to find all lines in the INVENTRY file that have a C or c in then, you enter:

```
FIND /i "C"  INVENTRY
```

This entry produces the following display:

```
---------- INVENTRY
Chair          2       Office Masters
Credenza       1       ABC Supply
Bookcase       9       Woodcare, Inc.
```

The FIND command finds both the uppercase C in Chair and Credenza and the lowercase c in bookcase, Woodcare and Inc.

Figure 14-4: Entries Matching Jackson in Several Files

```
C:\HSG>FIND "Jackson" WEEK9_1 WEEK9_8 WEEK9_15 WEEK9_22 WEEK9_29

---------- WEEK9_1
Bruce     Jackson      500

---------- WEEK9_8
Bruce     Jackson      525

---------- WEEK9_15
Bruce     Jackson      475

---------- WEEK9_22
Bruce     Jackson      520

---------- WEEK9_29
Bruce     Jackson      550

C:\HSG>
```

The record from each file is listed after the filename.

Displaying All Lines Except the Matches You negate the filter established with FIND when you use the /v switch. This means that all lines of the file except lines containing matching text will be returned. You might have an inventory file that contains discontinued items, back order items, current items in stock, and items that have been received but not yet stored. If you have the status of the records in each line you can look at all the entries except the discontinued items by searching for discontinued with the /V switch.

Suppose your file contains these entries and is called INVEN:. To locate all the records except discontinued items you enter the command like this:

```
FIND "Discontinued" INVEN
```

The result is all of the lines in the INVEN file that do not contain the word "Discontinued."

Adding Line Numbers to the Display With a long file, finding the lines that contain the text you want can be difficult. You can make it easier by including the line numbers in the output with the /n switch. You cannot combine this switch with the /c switch, since DOS will then ignore the /n switch. To use the /n switch, enter a command like:

```
FIND /n "ABC" INVENTRY
```

The output will include the line numbers in square brackets at the beginning of each line containing the specified text.

Sorting Output

The SORT command can reorder the lines in a file by the characters in a column of the lines. Each line in the file is considered a record, and columns are numbered beginning with 1 for the first column. The SORT command can process files as large as 64K. You can also sort information that you enter directly via the keyboard.

Output from the SORT command is written to the screen unless you direct it to a file or other device. To sort a file called TEXT by the entries in the first column and display the results on the screen, type:

```
SORT <TEXT
```

The sequence of the lines that display on your screen will depend on the contents of column 1. The numbers 0 through 9 in this position will display first and will be followed by A through Z. Letters are sorted by the language chosen, so if you have your computer set to use the conventions of a particular language, DOS will sort according to the alphabet's order for that language. DOS will look at the contents of columns to the right of column 1 after ordering the records by the entries in column 1. Assume that the file TEXT contains the following records:

```
C Williams    599.95
R Stevens     109.75
9 Parker      19.95
C Abbott      325.99
```

After using SORT, the records will display on the screen in this order:

```
9 Parker        19.95
C Abbott        325.99
C Williams      599.95
R Stevens       109.75
```

When a column contains the same entries on more than one line, DOS sorts those lines according to the entries in the next column to the right. This is why C Abbot is before C Williams. Since the first columns for these two lines have identical contents, DOS uses the second column to determine the order of the lines. DOS does not need to look at the other columns for the first and fourth lines, since these lines contain unique entries in the first column.

Sending the Sorted Output to a Location Other than the Screen You can use the redirection symbol (>) to send the output of SORT to another file or device. To write the same output to a file named FINALTXT, for example, change the command entry to:

```
SORT <TEXT > FINALTXT
```

To send the same sorted output to your printer, you would type:

```
SORT <TEXT > PRN
```

These two examples also show how you can combine the redirect input and output symbols in a single DOS command.

Reversing the Sort Order The SORT command is usually used to sort the lines in a file in ascending order, but you can tell DOS to sort in descending order by using the /r switch. When you use this switch, DOS sorts in reverse alphabetical order, followed by the numbers 9 to 0. For example, if you sort the TEXT file with this switch, your output will look like this:

```
R Stevens       109.75
C Williams      599.95
C Abbott        325.99
9 Parker        19.95
```

Beginning the Sort On a Different Column than the First You can choose any location within a line as the sorting column to control the order of the line. The /+n switch is used to indicate the offset of the column that you want to begin with. N is the number of the column where you want to start. The first column in the line is column 1. Thus, to sort on column 17 in the TEXT file, you would enter:

```
SORT <TEXT /+17.
```

The results would display on the screen like this:

```
B Parker        19.95
R Stevens       109.75
C Abbott        325.99
C Williams      599.95
```

To sort the lines beginning with the last name entries, you need to start sorting on column 3, like this:

```
SORT <TEXT /+3
```

Sorting Input You can also use the SORT command to sort entries that you make into a file. For example, if you need to enter orders into a new file, you can use the SORT command to enter these orders and sort them. An example of using the SORT command this way is:

```
SORT /+17 >TEMP.PRN
```

The top of Figure 14-5 shows several lines entered. When you press F6 or CTRL-Z and then press ENTER, DOS sorts the lines in the file you have entered. You can see how the SORT command has sorted the lines in the bottom half of Figure 14-5, which uses the TYPE command to display the contents of TEMP.PRN.

Limiting Output

The MORE command allows you to display one screen of information at a time, regardless of the amount of output the command you use will return to the screen. You can use MORE to limit the display of directory information on the screen, even though there is another alternative for limiting the directory listing with the /p switch, which displays one page at a time. The MORE command can also be used when the command does not have a special switch for limiting the display of output. You might use MORE with the TREE command or the TYPE command. If

you only want to use MORE to display one screen of a file at a time, you must enter the command like this:

```
MORE <TEXT
```

Figure 14-5: Sort Input and Output

```
C:\HSG>SORT /+17 >TEMP.PRN
Klineart Co      FG3409   Acct  500.98
Deaver Storage   AX7809   Mktg  125.67
Klineart Co      FG3405   Acct  125.85
Borris Studios   ST1234   Sale  435.17
Harris, Inc.     BY7211   Acct  675.99
^Z

C:\HSG>TYPE TEMP.PRN
Deaver Storage   AX7809   Mktg  125.67
Harris, Inc.     BY7211   Acct  675.99
Klineart Co      FG3405   Acct  125.85
Klineart Co      FG3409   Acct  500.98
Borris Studios   ST1234   Sale  435.17

C:\HSG>
```

The first command stores the sort output in TEMP.PRN. The second command displays the contents of TEMP.PRN.

This example displays one screen of text from the TEXT file at a time. When DOS fills the screen, it displays --More-- at the bottom of the screen. To see the next screen, press any key. To use the MORE command to display another command's output one screen at a time, DOS requires you to use the special DOS pipe character (|) to direct the command output to the MORE filter. To display only the first screen of lines from the file PAYABLE.TXT with TYPE, for example, you would enter this:

```
TYPE PAYABLE.TXT |MORE
```

Then when you press a key, DOS displays the next screen of data from PAYABLE.TXT. If you decide to end the display before DOS finishes, you can press CTRL-C to end the command without finishing. The MORE command creates a temporary file on the current drive, which it uses to store the information. It then deletes this temporary file when you are finished with the MORE command.

Building Pipes

Pipes are characters that redirect the output of one DOS command to be the input for another. This allows you to combine two or more DOS commands. You can also use the SORT and FIND commands to change the order of the lines or which of the lines in the files will be used. You can also combine the piping character with the characters for redirecting input and output.

Earlier you saw how you can use the piping character, the vertical bar (|), with the MORE command to list the output of a command one screen at a time. When you use the piping character, the information on the left of the vertical bar is the input to the command on the right side of the vertical bar. As an example, a command might look like this:

```
DIR A:|SORT /+10 >LISTING.PRN
```

This example uses the output created with the DIR A: command as the input for the SORT command. The >LISTING.PRN takes the output of the SORT command and stores it in a file called LISTING.PRN.

The pipe character is frequently used with the FIND, MORE and SORT commands to change the lines from a file the command uses. For example, you can use the FIND command with a pipe to only use those lines from the INVENTRY file that contain "ABC" by using a command like this:

```
TYPE INVENTRY |FIND "ABC" |SORT /+10
```

This command will use the contents of INVENTRY for the FIND command to produce only the lines that contain "ABC", which are sorted according to the values in the tenth column. This example produces these results:

```
Credenza        1       ABC Supply
Table           4       ABC Supply
```

At a Glance

In this chapter, you learned to use devices other than the standard devices associated with each command. You also learned to use the various filters that DOS provides for rearranging data or controlling the information displayed. The tasks that you can accomplish with the techniques in this chapter are:

- **Redirect output to another device—>**
- **Receive input from a non-standard device—<**

- **Use the sort filter to rearrange lines based on a specific column—SORT /+n**
- **SORT in reverse order—SORT /r**
- **Limit the display of data to one screen at a time—MORE**
- **Locate specific information in a text file—FIND**
- **Alter the input/output device — CTTY**
- **Use output from one command as the input for another (piping)—|**

Customizing DOS

DOS provides many customization features that alter the way DOS functions and affect the other programs you use. You can control the selected customization features and have an effect on the way your system will work. You can use several different methods for making customization changes to DOS. One option is to alter the CONFIG.SYS file which DOS reads every time you boot your system. DOS uses the CONFIG.SYS file to establish your system configuration. You can also customize DOS by creating an AUTOEXEC.BAT file that DOS also executes during the boot process. Since changes to CONFIG.SYS and AUTOEXEC.BAT are in effect every time you load the operating system, these methods are the most efficient way to make a consistent change to the operating system. Once your operating system is loaded you can still make customization changes by entering DOS commands at the DOS prompt. These changes will only be in effect for the current session.

What You'll Learn

This chapter provides examples of how you can customize DOS. After reading this chapter, you will be able to:

- **Change the appearance of the DOS prompt**
- **Create a CONFIG.SYS file to customize how DOS operates**
- **Create an AUTOEXEC.BAT file to be executed every time you load DOS**
- **Include device drivers as part of the configuration file**
- **Change how you use your disk drives**
- **Change the international configuration to use standards of other countries**

Changing the DOS Prompt

One of the simplest changes you can make to customize your system is changing the prompt that DOS displays when it is ready to accept a command. The default prompt display is the drive letter followed by a greater than symbol (>). If you have a hard disk, your prompt is likely to look like this: C>.

Using Special Characters to Change the Prompt

You will use the PROMPT command to change the prompt displayed. The entries you make after the PROMPT command determine the appearance of the DOS prompt on all subsequent command lines. You can enter the PROMPT command at the DOS prompt or include a PROMPT command in a batch file, such as AUTOEXEC.BAT.

To change the prompt, enter PROMPT, any text you want to include, and any of the special characters shown in Table 15-1.

Table 15-1: Special Entries for the Prompt Command

Characters	Displays	Example	Result
$q	=	Path $q $p	Path = C:\DOS
$$	$	Enter Command $$	Enter Command $
$t	Current time	Time: $t	Time: 10:57:00.00
$d	Current date	Date: $d	Date: Wed 02/20/91
$p	Current drive and path	$p -	C:\DOS -
$v	DOS version	vg	DOS Version 5.00>
$n	Current drive	Drive $n:	Drive C:
$g	>	pg	C:\DOS>
$l	<	lp$g	<C:\DOS>
$b	|	$t $b $n:$g	10:57:37.88 | C:>
$_	Advances cursor to the beginning of the next line	Enter command$_	Enter command _ (cursor is here)
$e	ASCII code 1B (escape) to be used with ANSI.SYS	$e[1m$pge[0m	C:\DOS>
$h	Backspace; to remove characters	thhhhh$h	10:57

The $p character provides important additional information for hard disk users since it causes DOS to display the complete path rather than just the drive designator. Although this does add the overhead of reading the current drive and path before displaying the DOS prompt, for hard disk users this may be worthwhile.

For floppy disks users who are not concerned with subdirectories this may not be the best approach—especially since each read will take a few additional seconds. Other options that may be of interest to users with specific applications are displaying the date and time. Since entries can be combined, you can actually choose to display both the date and time although the space used on each line severely limits the length of the DOS command that will display on the line.

Using Your Own Text as the Prompt

You can also use your own text for a prompt. When you enter your own text you do not need a dollar sign as in:

```
PROMPT What do you want?
```

Entering this command will change the prompt from the current setting to the question What do you want? If you enter the command as:

```
PROMPT What do you want?$g
```

the prompt will display as What do you want?> The > serves as a separator between the prompt and the DOS commands that you type.

Each time you enter the PROMPT command, you will replace the current setting with a new one. When you want to return the prompt to the default of the default drive name, enter PROMPT with nothing after it. Figure 15-1 shows several prompt commands and the results they create.

Figure 15-1: Examples of the PROMPT Command

```
C:\>prompt Time: $t$h$h$h$h$h$_Date: $d$_$p$g

Time: 18:55
Date: Wed 09-12-1990
C:\>prompt Current drive is drive $n:

Current drive is drive C:prompt

C>prompt You are using $v$_Enter your command$_$g

You are using MS-DOS Version 5.00.00
Enter your command
>
```

Each PROMPT command replaces the previous prompt with the entry after PROMPT. You can always return to the default by entering PROMPT with nothing after it.

Changing CONFIG.SYS

CONFIG.SYS is a file that tells DOS how you want it to work with various components of your system. It tells DOS which device drivers it should use with your hardware. The changes you can make to the way DOS works with entries in CONFIG.SYS cannot be made through entries at the DOS prompt. These changes are set up before the boot process has been completed.

The changes you can make with CONFIG.SYS will apply to your entire DOS session. In fact, because CONFIG.SYS is read during *every* boot procedure, these changes will be in effect for all sessions.

You can change the CONFIG.SYS file using the System Editor, EDLIN, or any word processor that creates ASCII files (files without the special information the word processor adds to indicate how the text is formatted). DOS has 13 commands you can use in the CONFIG.SYS file. These commands and examples are shown in Table 15-2. They are frequently used in combination yet each command is entered on a separate line in the file. If you expect to revise your CONFIG.SYS file frequently, you may want to include explanations in the file for the commands you have entered. You can add comments by entering REM at the beginning of the line and your comments afterwards. Some of the commands are grouped in special sections and are covered in more detail later in this chapter and in Chapter 16.

Table 15-2: Available Commands for CONFIG.SYS

Command	Use of Command	Example
BREAK	Sets when DOS checks for CTRL-C	BREAK=ON
BUFFERS	Sets number of buffers DOS uses to store data	BUFFERS=20
COUNTRY	Sets country conventions	COUNTRY=003,,C:\DOS\COUNTRY.SYS
DEVICE	Loads a device driver	DEVICE=ANSI.SYS
DEVICEHIGH	Loads a device driver in high memory	DEVICEHIGH=C:\DOS\RAMDRIVE.SYS
DOS	Selects where DOS is loaded and use of reserved memory	DOS=HIGH,UMB
DRIVPARM	Sets parameters of a disk drive	DRIVPARM=/D:4 /F:5 /N /S:32
FCBS	Sets the number of file control blocks (FCBs) that can be open	FCBS=5

Command	Use of Command	Example
FILES	Sets number of files that may be open at once	FILES=30
INSTALL	Installs and executes a program	INSTALL=C:\DOS\NLSFUNC.EX
LASTDRIVE	Sets the highest logical drive letter	LASTDRIVE=G
REM	Adds a comment	REM Set # of file to open
SHELL	Sets how DOS works with a command processor	SHELL=COMMAND.COM /P /E:1024
STACKS	Sets RAM used for processing hardware interrupts	STACKS=32,128

Setting When DOS Checks for CTRL-C

When you want to stop a DOS command, you can press CTRL-C. DOS is initially set to check if you press CTRL-C when it is waiting for keyboard input or when it is sending information to a printer. You can set DOS to also check if you have pressed CTRL-C when it is performing disk reads and writes by including this command in the CONFIG.SYS file:

```
BREAK=ON
```

This command can also be used from the DOS command prompt by entering:

```
BREAK ON
```

If you want to return the CTRL-C checking to the default setting, enter:

```
BREAK OFF
```

Finally, you can use the BREAK command to check the current break setting by entering BREAK with no text after it. This command will either display **BREAK is on** or **BREAK is off** to indicate the current setting.

Increasing the Number of Files DOS Can Open Concurrently

A file must be open before data can be read from or written to it. As a default, DOS can support as many as eight open files at one time.

When you work directly with DOS commands, your emphasis is frequently focused on one file at a time. Even if DOS has both an input and output file open as it copies data from one to the other you do not need to be concerned with this activity since fewer than eight files are open at one time. When you use many software applications, the program will be using a lot of different files to accomplish one activity, and all of them will need to remain open until the activity is complete. You may need to use more than eight files even when you think you are working with only one file. Using a word processor program as an example, you will have an open file for the text you are entering, one or more files the word processor uses for executing the program features, a backup file, as well as additional files such as dictionary and thesaurus files. If your program needs more than 8 files, it will not be able to open the additional files, and depending on the error recovery routines within your program, your program will display a message or lock your system.

Using the FILES Command to Increase the Number of Files Open Concurrently To allow for additional file usage, you must include a FILES command to tell DOS the maximum number of files that can be open at one time. You can check the documentation for your application software or enter FILES=20 in your CONFIG.SYS file as a good starting point for most other programs. The FILES command will support a number as high as 255. When you use the FILES command, DOS reserves space in RAM for a table that contains information on all open files.

Adding a BUFFERS Command to CONFIG.SYS You will also want to consider adding a BUFFERS command to allow more data from files to be read into memory —making access to needed information quicker. DOS uses buffers to hold file data as you read and write to the files. By increasing the number of buffers, you are increasing the temporary storage that DOS has to store the file information. By holding the information in a buffer, DOS performs fewer disk read and write operations to access file data. Although increasing the buffers makes programs run faster, it occupies memory that might be needed for other uses. When setting the number of buffers you must consider the trade-off between speeding up file access and increasing the buffers and decreasing the memory available for other software. Table 15-3 shows the default buffers based on the amount of RAM.

If DOS is stored in the high memory area (and the high memory area has enough room), DOS will store the buffers in the high memory area. This additional storage will provide more space in the traditional 640K area of RAM for

other applications. Each buffer requires approximately 532 bytes of memory. The number of buffers can be as high as 99. Although you can increase performance by increasing the number of buffers, performance does not increase beyond 50. The best performance is reached when you set the number of buffers to match the size of your hard disk. For hard disks less than 40 MB, the most efficient buffer size is 20. For hard disks between 40 MB and 79 MB, the most efficient buffer size is 30. For hard disks between 80 MB and 119 MB, the most efficient buffer size is 40. For hard disks with at least 120 MB, the most efficient buffer size is 50.

Table 15-3: RAM Used for Certain Number of Buffers

Available RAM	Number of Buffers	Amount of RAM Used by Buffers
Less than 128K	360 disk	2
More than 128K	360 disk	3
128K to 255K	5	2672
256K to 511K	10	5328
512K to 640K	15	7984

If you do not supply a BUFFERS command, DOS reserves space for the number of default buffers. Some applications require that buffers be greater than the default. When this occurs, use the higher number required by the software that you want to run. Also, many software applications will check the AUTOEXEC.BAT file to be sure that it provides for a sufficient number of buffers. Some disk-caching packages operate as efficient buffers and you will want to reserve less buffer space if you use one of them. To set the number of buffers, include a statement in the CONFIG.SYS file like:

```
BUFFERS=20
```

You can also use the BUFFERS command to specify a secondary cache to store temporary information in addition to the normal buffers. The secondary cache stores file information that comes after the file information stored in the primary buffers. The secondary cache assumes that the next block from the file you will need is the one that comes after the one in the primary buffer. While you are using the information in the first buffer, DOS copies the next part of the file into the secondary cache so it can be quickly copied to the primary buffer if you need it or dis-

carded if you do not. You can specify a secondary cache by separating the number of buffers and the secondary cache with a comma as in:

```
BUFFERS=40,8
```

The default size of the secondary cache is 1. 8 is the size you will normally want to use for the secondary cache. You do not want to use a secondary cache if you use a disk caching program such as the SMARTDrive disk-caching program included with DOS 5.0.

Setting Where DOS Is Loaded

By default DOS is loaded into the conventional memory of your computer. Depending on your system you might have another type of memory and may want to change the way that DOS is loaded. You can also change the way in which the reserved memory that DOS uses to load device drivers, which drive hardware components, becomes available for other applications. First you will want to make sure you understand the memory options that might be available on your PC.

Understanding the Different Types of Memory Your computer can have several different types of memory. By default, DOS is loaded into conventional memory. *Conventional memory* has a fixed location in your computer system. You can refer to the locations in conventional memory with addresses that begin with 0 and extend as high as 1 MB. Your system may have a full megabyte of memory or it may have less. Your programs are normally capable of using only memory addresses as high as 640K. Conventional memory above the 640K is referred to as *reserved memory.*

The other two types of memory are *extended memory* and *expanded memory.* Expanded memory was the first attempt at extending memory features and was made available on an expansion card. Its addresses were not fixed like conventional memory addresses, and in some cases its use was not as efficient as conventional memory because all locations used had to be mapped to conventional memory and required some conventional memory storage space to store addresses. Extended memory also has fixed address locations that start where conventional memory leaves off.

Controlling How DOS Is Loaded By default, DOS is loaded in conventional memory. If you have extended memory, you can load part of DOS into extended memory by including these two statements in your CONFIG.SYS file:

```
DEVICE=HIMEM.SYS
DOS=HIGH
```

The first statement loads the HIMEM device driver which manages extended memory. The second statement loads part of DOS into extended memory.

The other use of the DOS command in the CONFIG.SYS file is to make the extra memory in reserved memory available for other applications. Initially, you cannot use the unused reserved memory for other applications. If you have a 80386 or 80486 machine and want to use reserved memory for some device drivers and applications, you must tell DOS you want to have the memory available. To do this, use DEVICE commands to install HIMEM.SYS and EMM386.EXE and include this in your CONFIG.SYS file:

```
DOS=UMB
```

UMB stands for upper memory blocks, which is the name for the unused space of the reserved memory. You can combine this with the HIGH option as in:

```
DOS=UMB,HIGH
```

which loads DOS into extended memory and makes the unreserved space in the reserved memory available for use. In Chapter 16, you will see how you can use the LOADHIGH command to load programs into reserved memory. With the DOS command, if you want to explicitly use the defaults, you can use NOUMB in place of UMB to prevent DOS from letting you use reserved memory and you can use LOW in place of HIGH to load DOS entirely into conventional memory.

Loading Programs When You Start DOS

DOS includes some programs that you can enter at the command prompt or load when DOS reads the CONFIG.SYS file. These programs include FASTOPEN.EXE, KEYB.COM, NLSFUNC.EXE, and SHARE.EXE. FASTOPEN speeds file access time for frequently used files by tracking the location of these files. When you need a file you previously used, DOS will be able to determine its location from memory rather than from the directory on disk. KEYB and NLSFUNC are covered in more detail later in the chapter since they are used for customizing international settings. SHARE is needed when you use your computer in a multi-tasking or networking environment. Loading SHARE loads the code that supports file locking. *File locking* restricts a file's use to one user who wants to update the file. SHARE also loads code that supports *sharing*, which allows many users to read

the data in the file but prevents them from updating it. The switches used with SHARE allow you to allocate space for storing file sharing data and allow you to expand the number of files that can be locked at one time beyond the default of 20. You will find the parameters and switches these commands support in the command reference in Appendix F.

To load these commands when DOS reads CONFIG.SYS, you must use the INSTALL command. To load one of these programs, enter a command like:

```
INSTALL=KEYB.COM GR,,C:\DOS\KEYBOARD.SYS
```

This example loads the KEYB.COM file. The information shown after KEYB.COM are the parameters that the KEYB.COM file uses. The parameters are different for the four commands. These four commands remain in memory after they are executed, unlike commands such as FORMAT, which are removed from memory when they are finished.

Setting the Command Processor

COMMAND.COM is the file normally used as the command processor for DOS and it will process every command entry you make. You can create another command processor if you want to perform sophisticated customization. While the steps for modifying the processor are beyond the scope of this book, you can make changes to the command processor to change how the internal DOS commands work by renaming the commands. You can change the command processor used by entering a statement like this in the CONFIG.SYS file:

```
SHELL=C:\MY_PROC.COM
```

This command loads the command processor stored in the MY_PROC.COM file in the root directory. If the MY_PROC command processor uses switch settings, these entries can appear at the end of the SHELL command. If your command processor does not support switches, the Shell command will not make their use possible

You can also use the SHELL command to change how COMMAND.COM works by using the parameters available with COMMAND.COM. When you use the SHELL command with the COMMAND.COM command processor, you can use four different options. A sample SHELL command used with the COMMAND.COM command processor is:

```
SHELL=COMMAND.COM /E:256 /P
```

This loads the COMMAND.COM command processor. It assumes that this file is in the root directory since the path information is not included after COM-

MAND.COM. You can also set the command processor to use a different device for input and output by entering the input and output device name after COMMAND and any path information. The /E: selects a number from 160 to 32,768 bytes—the operating system uses this number for storing environment variables. An *environment variable* is a variable name and variable value that you can set and reference using DOS. The /P makes the new processor permanent, which means you cannot return to the prior version of the command processor by typing EXIT. The /P parameter also tells DOS to execute any AUTOEXEC.BAT file. The COMMAND.COM file has another parameter, /C, which you can follow with a string. When you use the COMMAND.COM command with a /C parameter, DOS performs the command described by the string and returns to the previous command processor. This means the string must be an acceptable command that you would be able to enter directly into the command prompt.

- ***NOTE:*** If you plan to create and use several environment variables, you may need to use the SHELL command to insure that DOS has set aside enough memory to store the environment variables you want.

Setting the Number of File Control Blocks

Some older software uses file control blocks (FCB) to provide a program that has the ability to keep track of important information for a file. DOS can use FCBs or file handles. Recent applications use file handles, so it is only if you are using an older application that you may need to change the number of FCBs you can use concurrently. DOS defaults to 4, but you can change it by entering FCBS= and a number between 1 and 255 on a line in the CONFIG.SYS file. You should only change the setting for file control blocks when you are using an application in which the default setting is insufficient.

Setting the Memory DOS Uses for Hardware Interrupts

Every time your computer receives a hardware interrupt, such as pressing CTRL-BREAK, it tries to process the hardware interrupt. If the computer receives more than one hardware interrupt, it tries to put information for each unprocessed hardware interrupt into a stack. By default, DOS creates nine stacks of 128 bytes each to store information about hardware interrupts until it is ready to be processed for all computers except the old PC's, XT's, and PC-Portables, which all do not reserve space for storing multiple hardware interrupts. You can change the number and size of the stacks with the STACKS command in the CONFIG.SYS file. To make this change add enter a command like:

```
STACKS=32,128
```

This command reserves 32 stacks of 128 bytes to process hardware interrupts. This allows the computer to handle more hardware interrupts. Changing the stack was mostly used in prior releases of DOS since too many hardware interrupts would cause the computer to lock up when the stack area DOS reserved for handling hardware interrupts became overfilled.

Changing AUTOEXEC.BAT

In Chapter 12 you learned that batch files allow you to provide a series of DOS commands which can be executed easily because they are stored in a file. AUTOEXEC.BAT is a special batch file that DOS runs *every time you load the computer*. You can put instructions in this file for setting the current directory, DOS prompt, checking the date and time, and path where DOS looks for files. You can include any DOS commands you like in the AUTOEXEC.BAT file. The AUTOEXEC.BAT file is used to store DOS commands that you want executed every time you start DOS. When you use the DOS 5 install program, it will place a command in the AUTOEXEC.BAT file that automatically starts the DOS 5 Shell for you. The ability to enter commands in AUTOEXEC.BAT makes customizing DOS easier since you only enter the commands into a batch file once and do not have to enter them again.

Creating an AUTOEXEC.BAT File

AUTOEXEC.BAT files are normally not lengthy. You might find that using COPY CON AUTOEXEC.BAT is a good, quick way to create one if there is not already one on your system. Although this approach does not allow you to make modifications, it is adequate if you only want a few instructions in the file.

If you need to create a lengthy file or modify an existing file you will want to create an ASCII file with your word processor, or use the DOS System Editor. Instructions for using the DOS system editor as well as details on some of the commands used in batch files are found in Chapters 11 and 12.

Examining a Sample AUTOEXEC.BAT File

You can put any information you want into a DOS batch file. One of the first tasks you normally want the batch file to complete is to stop echoing the commands that are processed to the screen. You would enter ECHO OFF to achieve this effect. If your system does not automatically remember the time and date, you may want to add the DATE and TIME commands to this file. Other tasks that are often handled with the batch file are setting the path, changing directories, and starting a menu or a specific program. The following lines might be the contents of one such file:

```
TIME
DATE
PROMPT $P$G
DOSSHELL
```

This AUTOEXEC.BAT file prompts for a time and date and then starts the DOS Shell program. A more advanced AUTOEXEC.BAT file may look like this:

```
ECHO OFF
PATH C:\DOS;C:\WORD;C:\123
SUBST E: C:\123\FINANCE\BUDGET
PROMPT Enter the number of your menu selection -
MENU1
```

The first command in this AUTOEXEC.BAT file stops DOS from displaying the commands in this file. The next two commands set the path and use drive E to refer to the C:\123\FINANCE\BUDGET subdirectory. The PROMPT command changes the appearance of the command prompt. Finally, this AUTOEXEC.BAT file executes the executable program MENU1. This MENU1 program would probably prompt the user to enter a number or letter to select the application they want to work with.

Changing Drive Assignments

DOS has three changes you can make to disk drives. These changes can affect logical drives, add physical drives, and change the parameters physical drives use. You can change the number of logical disk drives so you have more letters to refer to your physical drives or more letters to use with the SUBST command. You can add additional disk drives when you have more than two floppy disk drives. You can change the settings that disk drives use.

You can change the number of logical drives to use more letters to refer to your physical disk drives. (DOS uses letters to identify each drive.) You will want to increase the number of logical drives when you add another floppy disk drive, or when you want to use more than one drive letter for a physical drive. For example, if you have two floppy disk drives that are different formats, you may want to use two letters each for the disk drive when you want to copy files between floppy disks of the same format. To set the maximum number of logical drives, enter a line like this in the CONFIG.SYS file:

```
LASTDRIVE=N
```

This command reserves the space in memory that DOS uses to create a table that lists the logical drive letters and the physical drives. Since the table that this command establishes occupies the computer's memory, do not use a higher letter than you expect to use. This particular example lets you use the letters A through N for your disk drives. A and B are automatically reserved for floppy disk drives. If you do not use this command, the table DOS creates is set to use logical drives A through E. Using the ASSIGN and SUBST commands, you can assign other letters to the physical drives.

Another change you may want to make is when you add another floppy disk drive. DOS assumes you have two floppy disk drives that are labeled A and B. If you only have one floppy disk drive, DOS assigns both logical drive letters to the same floppy disk drive. If you have more than two floppy disk drives, you must load a device driver that tells DOS about the extra floppy disk drives. This is done by loading the DRIVER.SYS device driver. As described later, you must use the DEVICE command in your CONFIG.SYS file to load a device driver. To use the DEVICE command with the DRIVER.SYS device driver, you must include a command like this:

```
DEVICE=DRIVER.SYS /D:2 /C /F:0 /H:2 /S:9 /T:80
```

This command loads the device driver named DRIVER.SYS. The /D: followed by a number between 0 and 127 assigns a physical drive number to the disk drive. 0 is automatically assigned for drive A and 1 is automatically assigned for drive B, even if it is not present. The /C tells DOS that the disk drive can detect that the drive door is not shut. This is omitted if the disk drive cannot tell whether the drive door is shut. The /F: is followed by 0, 1, 2, or 7 to indicate the storage capacity of the disks. The /H: is followed by the number of heads the drive has. This optional argument only needs to be supplied if you do not want to use the default of 2. The /S: is followed by the number of sectors on each track of the disk drive. This optional argument only needs to be supplied if you do not want to use the default of 9. The /T: is followed by the number of tracts the drive has. This optional argument only needs to be supplied if you do not want to use the default of 80. The different commands you can enter for the different drive types are (assumes that it is the third floppy disk drive and the drive can detect if the door is open):

160K	DEVICE=DRIVER.SYS /D:2 /C /F:0 /H:1 /S:8 /T:40
180K	DEVICE=DRIVER.SYS /D:2 /C /F:0 /H:1 /T:40
320K	DEVICE=DRIVER.SYS /D:2 /C /F:0 /S:8 /T:40
360K	DEVICE=DRIVER.SYS /D:2 /C /F:0 /T:40

1.2 MB (5 1/4")	DEVICE=DRIVER.SYS /D:2 /C /F:1 /S:15
720K (3 1/2")	DEVICE=DRIVER.SYS /D:2 /C /F:2
1.44 MB (3 1/2")	DEVICE=DRIVER.SYS /D:2 /C /F:7 /S:18
2.88 MB (3 1/2")	DEVICE=DRIVER.SYS /D:2 /C /F:7 /S:36

The third change you may want to make in the CONFIG.SYS file for disk drives is changing the properties DOS assumes about physical disk drives. A sample of this command is:

```
DRIVPARM=/D:4 /F:5 /N /S:32 /T:32
```

This example defines the fifth physical device as a non-removable hard disk that has 32 sectors per track and 32 tracks per side. Like using DRIVER=DRIVER.SYS, you can use the /D, /C, /F, /H, /S, and /T parameters. The /F parameter can also include 5 for a hard disk drive and 6 for a tape drive. As shown in the example, the /N is used to indicate that the device you are changing the parameters for is nonremovable. You can also use a /I parameter after the /H to indicate that the drive you are defining is a 3 1/2 inch floppy disk drive that is not supported by your system. This parameter is used for some IBM PC/AT compatibles when the ROM-BIOS does not support 3 1/2 inch disks.

■ ***NOTE:*** You cannot use DRIVPARM to change the parameters of a disk drive to parameters the disk drive cannot use. For example, you cannot use this command to change a 360K disk drive to read high density 1.2 MB 5 1/4 inch diskettes.

Assigning Logical Drives

Once you have the number of logical drives you want, you may want to assign these logical drives to physical drives. You can assign logical drives to network drives, extended DOS partitions, RAM drives, or another drive or directory. You can also change the properties of drive A or B by assigning another logical drive to the physical drive with the properties you want.

If you want to assign two logical drive letters to a disk drive, you can make the change from the DOS command prompt or in the CONFIG.SYS file. To make the change from the command prompt or in a batch file, use the command SUBST. In Chapter 8 you used this command to assign a path to a logical drive although it can also be used to change drive assignments. An entry of SUBST E: C:\ will allow you to refer to drive C with either C: or E:. To assign two logical drive letters to a physical disk drive using CONFIG.SYS, include a command like:

```
DEVICE=C:\DOS\DRIVER.SYS /D:0 /F:1
```

This assigns another drive letter to drive A, with a physical disk drive number of 0 that is a 360K disk drive.

If you change your disk drive to a different type of disk drive and your computer does not recognize the change, DOS will display errors when you try to use the drive. One solution is to load the device driver that accompanies the disk drive. Another solution is to assign a different logical drive to the physical disk drive and select the correct properties for the second logical drive assigned to the physical drive. For example, suppose you have switched drive B from a 5 1/4 inch 360K disk drive to a 3 1/2 inch 1.44 MB disk drive. To use a logical disk drive letter for this new disk drive, you would include this statement in your CONFIG.SYS file:

```
DEVICE=C:\DOS\DRIVER.SYS /D:1 /F:7 /S:36
```

Assuming the hard disk is drive C, you can now use B and D (the next unused logical drive letters) to refer to the second floppy disk drive. When you use B, the computer will display error messages, but you can use D without problems. Adding logical drives this way does not require a LASTDRIVE command.

Adding Device Drivers

One of the features of DOS is that it can work with different hardware. For each attachment to the computer, like a disk drive or monitor, DOS automatically includes device drivers for the keyboard, monitor, hard and floppy disk drives, and ports such as LPT1. DOS also includes other device drivers that you can load if you want to use the feature the device driver provides. You have already seen some DEVICE commands used to load device drivers for cases where you want to change CONFIG.SYS. Later, you will see how you can use DISPLAY.SYS and PRINTER.SYS when you want to set your display or printer to use the standards of other countries. Device drivers also provide other features for how you use your computer system. You can use device drivers to take advantage of extended or expanded memory, create an additional drive, and create a disk cache so file operations perform faster. In addition, you can use the ANSI.SYS device driver which changes how you can display graphics, control the cell pointer movement, and reassign keys.

- ***NOTE:*** Do not use DEVICE=COUNTRY.SYS or DEVICE=KEYBOARD.SYS since either command will lock up your system. DOS automatically loads these device drives when it needs them.

Loading a Device Driver

The DEVICE command is used in the CONFIG.SYS file to load device drivers. You can use this command to load drivers provided with DOS, as well as those that are provided with hardware devices such as a mouse. You might enter DEVICE=RAMDRIVE.SYS to load the device that establishes a virtual drive in memory.

Additional parameters can be used with the DEVICE command if the device driver accepts switch entries. Since the default size for a virtual disk is 64K, you will only need to specify this parameter if you want to specify a different size. To create a virtual disk that is 128K you would enter

```
DEVICE=C:\DOS\RAMDRIVE.SYS  128.
```

Loading a Device Driver into Reserved Memory

As you saw with the command DOS=UMB which you can include in the CONFIG.SYS file, you can make part of the reserved memory available for device drivers and other applications. To load a device driver into reserved memory you use the DEVICEHIGH command in place of DEVICE. Some device drivers can be loaded into reserved memory and some cannot. To load a device driver into reserved memory, the command would look like this in the CONFIG.SYS file:

```
DEVICEHIGH=C:\DOS\RAMDRIVE.SYS  /e
```

or:

```
DEVICEHIGH SIZE=28000 C:\DOS\RAMDRIVE.SYS  /e
```

The difference between the two commands is that the second command checks the amount of reserved memory available for loading device drivers and other applications. Assuming the number of available bytes is larger than the hexadecimal number after the equals sign, DOS loads the device driver in reserved memory. Otherwise, DOS loads the device driver in conventional memory. The first command loads the device driver in reserved memory. If DOS is unsuccessful, your system may hang. Using DEVICEHIGH requires that you use DOS=UMB and DEVICE=EMM386.EXE, which is covered in Chapter 16.

■ ***NOTE:*** You can find the hexadecimal value you want for the SIZE parameter of the DEVICEHIGH command by entering MEM/PROGRAM when the device driver is loaded into low memory. This lists the loaded programs, their locations, and sizes.

Using ANSI.SYS

ANSI.SYS is a device driver that you can use to display text using different colors and attributes, control cursor movement, and change the function assigned to keys. ANSI.SYS provides these features so you have better control over the screen's appearance and so you can change the characters assigned to any key.

To use ANSI.SYS, you must load the ANSI.SYS device driver. Once loaded, you use ANSI.SYS by entering escape sequences. All ANSI escape sequences are an escape character and an open bracket ([ ). Most of these examples display the ESC character as ^[since in most word processors this is how it will appear when you use a file with ESC characters. You can also enter the ESC character with the PROMPT command by using $e. To enter the ESC character in EDLIN, press CTRL-V and type a [. To enter the ESC character in Microsoft Word, enter ALT-27. Most word processors let you enter ESC characters directly into a document but you must check the documentation for how you do it with your specific package. ANSI escape sequences can be entered directly into the command prompt, be included in a batch file, or be used by a program. For example, you can use ANSI escape sequences as part of a menu program where you are using them to set the color, attributes and position of the text on the menu screen. Unlike most DOS commands, the letter at the end of the ANSI escape sequence is case sensitive.

Loading ANSI.SYS You can load ANSI.SYS by adding a line to your CONFIG.SYS file that looks like this:

```
DEVICE=C:\DOS\ANSI.SYS /X /L
```

This DEVICE command loads the ANSI.SYS device driver that is in C:\DOS. The /X parameter tells DOS you are using a keyboard that has two sets of cursor movement keys and 12 function keys. The original keyboards that you will not use this parameter with has only one set of cursor movement keys and ten function keys. The /L parameter tells ANSI.SYS that when you are using another application that alters the number of rows on the screen, ANSI.SYS will prevent the program from returning to the default of 25 when you exit the program. The /X and /L are optional and since they are a DOS command they are not case sensitive.

Using ANSI.SYS to Change the Screen's Appearance You can use ANSI.SYS to change the appearance of batch files as well as to incorporate ANSI.SYS sequences in other programs so the program can use position output on the screen and move the cell pointer to where input is required. The ANSI.SYS sequences to move the cursor are listed in Table 15-4. Once you move the cursor to a new position, you

can enter text which appears at the new location. You can use ANSI escape sequences to position text as well as set the colors and text attributes.

Table 15-4: ANSI.SYS Codes for Cursor Movement

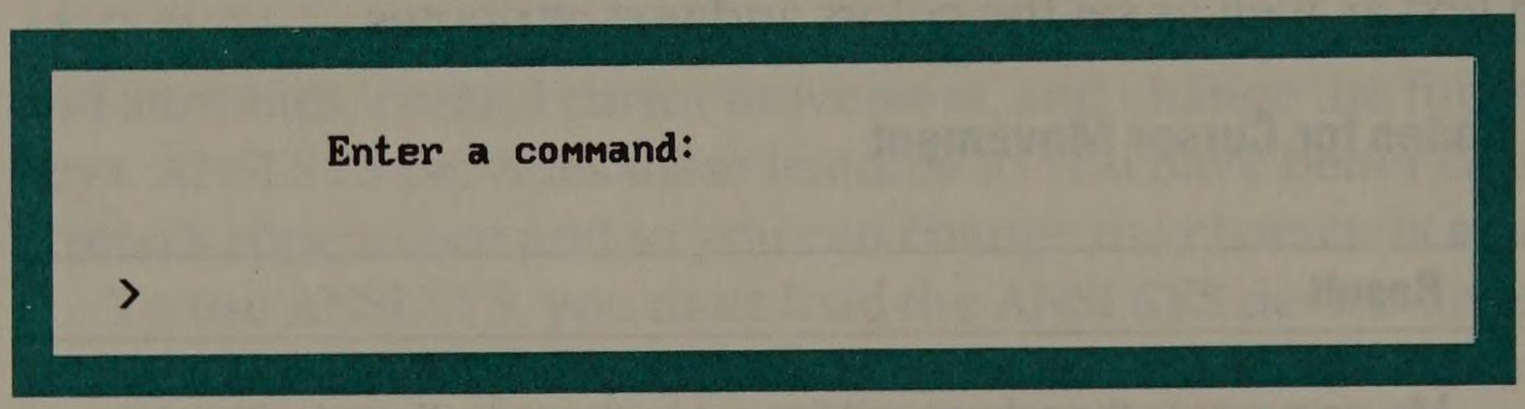
ANSI.SYS Codes	Result
ESC[*line*;*column*H	Moves cursor to the column and line set by the code. If used without line and column, moves the cell pointer to the upper left corner of the screen.
ESC[*line*;*column*f	Moves cursor to the column and line set by the code. If used without line and column, moves the cell pointer to the upper left corner of the screen.
ESC[*# of lines* A	Moves cursor up the number of lines.
ESC[*# of lines* B	Moves the cursor down the number of lines.
ESC[*# of columns* C	Moves cursor to the right of the number of columns.
ESC[*# of columns* D	Moves cursor to the left of the number of columns.
ESC[6n	Produces a cursor position report.
ESC[s	Remembers current cursor position to return to it at a later point.
ESC[u	Returns cursor to position saved with ESC[s.
ESC[2J	Erases the screen and moves the cursor to the upper left corner.
ESC[K	Erases the characters to the right of the cursor's position.

As an example, you can enter this command at the command prompt to use the ANSI escape sequences to position the prompt and the cursor:

```
PROMPT $e[2J$e[2;10fEnter a command:$e[5;0f$g
```

This example changes the prompt. This PROMPT command uses the $e to enter the ESC character. As this example shows, ANSI sequences are combined one after another with no spaces separating them. In the example above, the first escape sequence clears the screen. The second one moves the cursor to the third row and eleventh column (since both column and row numbers start with 0). Next, the prompt displays the text **Enter a command:** and then the last ANSI sequence moves the cursor to the sixth row and the first column before displaying > as the prompt. After entering this command, the command prompt looks like the one shown in Figure 15-2.

Figure 15-2: Command Prompt Changed with ANSI Escape Sequences



This command prompt accepts entries just like the default command prompt does.

Another option you have with ANSI escape sequences is to change the color and other attributes of the text. Text attributes are set by entering ^[[, the graphic attribute numbers separated by semicolons and an m, as in ^]]31;5m.You can use the colors listed in Table 15-5 to change the color of the text or the color behind the text. These color changes let you customize your screen. You can also add attributes like boldface (1), underscore for monochrome monitors (4), blinking (5), reverse video (7), and hidden (8). These attributes are included by using an ANSI escape sequence such as ^[[1m which boldfaces all text that appears after the escape sequence. When you want to return the attributes to the default, use ^[[not]]0m. You can combine the attributes and colors as in ^[[5;30;41m, which will flash the subsequent text with black letters on a red background. An example using the ECHO command in a batch file is:

```
@ECHO ^[[2J
@ECHO ^[[1;10f^[[5;32;47mHello^[[0m
```

This example clears the screen and then displays **Hello** starting at the second row and eleventh column. The text is blinking green letters on white text. When these lines are in a batch file the @ before ECHO means that the word echo will not appear on the screen when you execute the batch file. With the second command, if you separated the ANSI escape sequences onto different lines, DOS would advance to the beginning of the next line before ANSI.SYS performed the next escape sequence.

Table 15-5: Colors Added with ANSI.SYS Codes

Colors	Foreground (Color of Text)	Background (Color Behind Text)
Black	30	40
Red	31	41
Green	32	42
Yellow	33	43

Colors	Foreground (Color of Text)	Background (Color Behind Text)
Blue	34	44
Magenta	35	45
Cyan	36	46
White	37	47

The other screen change you can make with ANSI escape sequences is changing the mode. While you can use the MODE command to do the same function, you can use the ANSI escape sequences in the middle of your own programs. To set the mode, use the escape sequence ^[[=#h. The # represents a number between 0 and 19. The different numbers have the screen type settings listed in Table 15-6. As an example, you can change from 80 X 25 color to 40 X 25 color by using the escape sequence ^[[=1h. After entering the escape sequence, the screen display will use 40 columns instead of 80 and the characters will be wider.

Table 15-6: Mode Settings Available with ANSI Escape Sequences

Mode Number	Screen Width and Type
0	40 X 25 monochrome
1	40 X 25 color
2	80 X 25 monochrome
3	80 X 25 color
4	320 X 200 color graphics
5	320 X 200 monochrome graphics
6	640 X 200 monochrome graphics
7	Wraps text at the end of the line
14	640 X 200 16 colors
15	640 X 350 monochrome
16	640 X 350 16 colors
17	640 X 480 2 colors
18	640 X 480 16 colors
19	320 X 200 256 colors

Using ANSI.SYS to Reassign Keys Another feature ANSI.SYS provides is reassigning keys. You can use reassignments to make extended characters you use frequently more accessible. You can also assign multiple characters to a key so by pressing a key, you have several DOS commands entered for you. As an example, suppose you want to type the British pound symbol when you type SHIFT-4 which normally produces a $. You can enter an ANSI escape sequence that looks like ^[[36;156p or ^[["$";"£"p. In both examples, the ^[represents the ESC character which displays as ^[when you look at a file containing this character. The first number after the ^[[telling DOS that this is an ANSI escape sequence is the number of the key or the character in quotes that you want to change. These keys are listed in Tables 15-7 and 15-8. After the semicolon is the character or characters you want to use for this key. Finally, the escape sequence concludes with a p to indicate to ANSI.SYS that this escape sequence is for a key reassignment. As the two examples show, you can either supply the ASCII code or you can enter the character in quotes.

Table 15-7: Keys and Codes for Letters to Use for Key Reassignments

KEY	Lowercase	Uppercase	With CTRL	With ALT
A	97	65	1	0;30
B	98	66	2	0;48
C	99	67	3	0;46
D	100	68	4	0;32
E	101	69	5	0;18
F	102	70	6	0;33
G	103	71	7	0;34
H	104	72	8	0;35
I	105	73	9	0;23
J	106	74	10	0;36
K	107	75	11	0;37
L	108	76	12	0;38
M	109	77	13	0;50
N	110	78	14	0;49
O	111	79	15	0;24

KEY	Lowercase	Uppercase	With CTRL	With ALT
P	112	80	16	0;25
Q	113	81	17	0;16
R	114	82	18	0;19
S	115	83	19	0;31
T	116	84	20	0;20
U	117	85	21	0;22
V	118	86	22	0;49
W	119	87	23	0;17
X	120	88	24	0;45
Y	121	89	25	0;21
Z	122	90	26	0;44

Another possible ANSI escape sequence you may want to use is ^[[0;123;"dollars"p. This escape sequence uses the first two numbers to identify the key. Keys combined with ALT, the function keys and some of the cursor movement keys have two numbers, a 0 and another number. When ANSI.SYS processes this escape sequence, it realizes the key you are redefining uses two numbers since the first number is 0. Every ASCII code or string in quotes after the key is assigned to the key you are changing. In this example, DOS will enter the word dollars every time you press ALT-4.

Table 15-8: Keys and Codes for Other Keys for Reassignment

Keys	Alone	With SHIFT	With CTRL	With ALT
0	48	41	N/A	0;129
1	49	33	N/A	0;120
2	50	64	N/A	0;121
3	51	35	N/A	0;122
4	52	36	N/A	0;123
5	53	37	N/A	0;124
6	54	94	N/A	0;125
7	55	38	N/A	0;126

Keys	Alone	With SHIFT	With CTRL	With ALT
8	56	42	N/A	0;127
9	57	40	N/A	0;128
-	45	95	N/A	0;130
=	61	43	N/A	0;131
F1	0;59	0;84	0;94	0;104
F2	0;60	0;85	0;95	0;105
F3	0;61	0;86	0;96	0;106
F4	0;62	0;87	0;97	0;107
F5	0;63	0;88	0;98	0;108
F6	0;64	0;89	0;99	0;109
F7	0;65	0;90	0;100	0;110
F8	0;66	0;91	0;101	0;111
F9	0;67	0;92	0;102	0;112
F10	0;68	0;93	0;103	0;113
F11	0;133	0;135	0;137	0;139
F12	0;134	0;136	0;138	0;140
HOME	0;71	55	0;119	N/A
END	0;79	49	0;117	N/A
UP ARROW	0;72	56	N/A	N/A
DOWN ARROW	0;80	50	N/A	N/A
LEFT ARROW	0;75	52	0;115	N/A
RIGHT ARROW	0;77	54	0;116	N/A
PAGE UP	0;73	57	0;132	N/A
PAGE DOWN	0;81	51	0;118	N/A
INSERT	0;82	48	N/A	N/A
DELETE	0;83	46	N/A	N/A
TAB	9	0;15	N/A	N/A
PRINT SCREEN	N/A	N/A	0;114	N/A
NULL	0;3	N/A	N/A	N/A

With some keys you must use the ASCII value. As an example, suppose you want DOS to switch to the WorkPerfect directory called WP51 and start WordPerfect with the file DAY_PLAN every time you press F11. The ANSI escape sequence in a batch file would look like:

```
echo  ^[[0;133;"CD\WP51";13;"WP DAY_PLAN";13p
```

The 0;133 is the code for F11, which is a function key available with the enhanced keyboards. Everything after 0;133; is the entry DOS will provide whenever you press F11. In this example, DOS types CD\WP51 to switch to the WP51 directory. The 13 is the ASCII code for ENTER, so DOS processes this command. Next, DOS enters WP DAY_PLAN which loads the WordPerfect program and the file called DAY_PLAN. The final 13 tells DOS to process this command. The p tells ANSI.SYS that this is a key reassignment. You could use this in a file like AUTOEXEC.BAT so you make all of your key reassignments at the beginning of your session. Besides using the ASCII codes for the ENTER key, you may also need to use 34 when you want a double quote since ANSI.SYS treats double quotes as the beginning of a string.

You can return keys to their original setting by using the code or string assigned to the key for both keys described, as in ^[[36;36p or ^[[0;133;0;133p. Most of the time you will use the key code to reset the key since the character assigned to the key is also used by another letter.

■ ***NOTE:*** Many software applications have their own key assignments which are not affected by reassigning keys with ANSI.SYS.

Changing the International Configuration

You can change the country that DOS assumes you are using. The country DOS assumes you are using sets the language conventions, keyboards, and character sets. These different features vary by country since different languages use unique characters and countries use different standards for the format of dates and times, currency symbol and character sort order. For example, if you are typing several Spanish documents, you may want frequently used Spanish letters that are not used in English to be easily available with a keyboard entry. You can tell DOS the keyboard you are using and change other DOS settings to match foreign standards. These changes are made in the CONFIG.SYS and AUTOEXEC.BAT files. DOS 5.0 can use the language conventions, keyboards, and character sets for most European countries. By default, DOS is set to use United States conventions although if you selected another country during installation, DOS will have added

the proper commands to the CONFIG.SYS and AUTOEXEC.BAT files for you. When you set DOS to match your country, you are changing the character set that appears, selecting the keyboard you are using, and the conventions that you want to follow.

Setting the Character Set with Code Pages

Your computer has 256 codes it can use to represent characters. DOS uses code pages to match the codes with 256 characters. These code pages are the feature that allow a code to represent an A umlaut (two dots above it) in one language and an A with a tilde (~) above it in another. By changing the active code page, you can change the characters that DOS uses. For each of the code pages, the characters with the codes between 0 and 127 are the same. These are the standard character set. The characters with codes above 127 are the extended character set. These extended characters vary by code page to offer different characters. Code pages are assigned different numbers. The standard for the United States is code page 437, although code page 850 is also frequently used. Appendix E lists the different code pages and the extended characters that are available with each one.

DOS has two types of code pages. *Hardware code pages* are internal to DOS. The other type of code pages are *prepared code pages* which are provided by software. Some systems such as CGA and monochrome screens cannot use prepared code pages.

You can check the current code page by entering **CHCP** and pressing ENTER. When you enter this command, DOS displays the active code page and any prepared system code pages.

Changing Code Pages The steps you perform to change a code page depend on whether you want to use a hardware code page or a prepared code page. To use a hardware code page, enter:

```
CHCP code page Number
```

To use a prepared code page for your monitor, you must first add one or more DEVICE commands to CONFIG.SYS to tell DOS it should reserve part of RAM to store a prepared page. This is done for the monitor and possibly the printer. Some IBM printers can have their code pages changed with the DEVICE command. Many printers can change their code pages if you send special codes to the printer. To reserve RAM space for a prepared code page for a monitor, the DEVICE command in the CONFIG.SYS file may look like this:

```
DEVICE=DISPLAY.SYS CON=(EGA,hardware code page number,1,1)
```

To reserve RAM space for a prepared code page for a printer, your DEVICE command in the CONFIG.SYS file may look like this:

```
DEVICE=PRINTER.SYS LPT1=(4201,code page number,1)
```

If DISPLAY.SYS or PRINTER.SYS is not in the root directory, you must include the path information, as in C:\DOS\DISPLAY.SYS. The CON, LPT1, LPT2, or LPT3 instructs DOS that you are setting the code page for your monitor or a printer. After the CON or LPT1 are the parameters the device uses. For the console or monitor, you must supply the display adapter you have which is either EGA for EGA or VGA monitors, or LCD for LCD monitors. For an IBM printer, this is the number 4201, 4208, or 5202 representing the IBM printer model that will use the new code page. Many printers have their own code pages which you can select by sending special codes to the printer. The code page number is the number your system uses. By providing this optional information, you can later switch back to the code page. The number after the code page number is the number of prepared code pages that you want to load. For EGA and VGA monitors, this number can be 0, 1, or 2. For LCD monitors, this number can be 0 or 1. For printers, this number depends on the printer you are using. If you omit this optional argument, DOS assumes it is 0. The fourth number (used only for monitors) is the number of fonts supported for each code page. If this number is omitted, DOS assumes it is 0. Since this command is in the CONFIG.SYS file, you must reboot the computer for the change to take effect.

After you have these DEVICE commands, you must load the NLSFUNC program by including NLSFUNC in the AUTOEXEC.BAT file, the line INSTALL=C:\DOS\NLSFUNC.EXE in the CONFIG.SYS file, or by entering NLSFUNC at the DOS command prompt. This loads the NLSFUNC program which lets DOS load and switch between prepared code pages.

Once the DEVICE command is in the CONFIG.SYS file and the NLSFUNC program is loaded, you can load the code page. To load a code page use the MODE CODE PAGE PREPARE command which you can abbreviate as MODE CP PREP. This command is either included in the AUTOEXEC.BAT file or is entered directly at the DOS command prompt. To load the code page enter one of the commands from Table 15-9. These commands assume you want to load page 850 and the printer is connected to LPT1:.

When you use one of these commands, you will want to substitute the code page number for the code page you want and the printer connection you want in place of LPT1. These examples assume that the code pages files with the .CPI extensions are in the C:\DOS directory. You must use separate commands for

each device that you want to set the code page. If you want to load more than one code page for a device, enter both code page numbers separated by a space in the parentheses, as in MODE CON CP PREP=((850 863)C:\DOS\EGA.CPI). This requires that the DEVICE command in the CONFIG.SYS file has reserved room for more than one code page.

Table 15-9: Commands to Load Code Pages

Device to Use New Code Page	Command to Enter
EGA or VGA monitor	MODE CON CP PREP=((850)C:\DOS\EGA.CPI)
LCD monitor	MODE CON CP PREP=((850)C:\DOS\LCD.CPI)
IBM Proprinter Model 4201	MODE LPT1 CP PREP=((850)C:\DOS\4201.CPI)
IBM Proprinter Model 4207 or 4208 or the X24 or XL24	MODE LPT1 CP PREP=((850)C:\DOS\4208.CPI)
IBM Quietwriter III Printer model 5202	MODE LPT1 CP PREP=((850)C:\DOS\5202.CPI)

Finally, with the code pages loaded, you can switch code pages for all devices using the CHCP command by entering CHCP and the code page number as in CHCP 850. If you only want to set the code page for one device, use the MODE CODE PAGE SELECT command which you can abbreviate as MODE CP select. The command may look like this:

```
MODE CON CP SELECT=850
```

This example changes the code page for the console to 850. Assuming the DEVICE statement has reserved room for the code pages, the NLSFUNC program is loaded, the code page is loaded, the hardware supports using another code page, and the code page selected is supported for the country keyboard selected. The last condition is discussed later with the keyboards. If somewhere along the process of changing the code pages, DOS cannot process the command, DOS will display a message indicating the problem. An example of this will occur if you try to load a prepared code page on a system with a CGA monitor. DOS will display a message since these monitors cannot support prepared code pages. For this reason, you will want to check the valid code pages for the language you want to use which are listed in the second column of Table 15-10.

Table 15-10: Codes to Use for Other Countries

Country/Language	Valid Code Pages	Keyboard Code	Country Code
Arabic	437		785
Belgium	437 and 850	be	032
Denmark	865 and 850	df	045
English (International)	437 and 850		061
Finland	437 and 850	su	358
France	437 and 850	fr	033
French Canadian	863 and 850	cf	002
Germany	437 and 850	gr	049
Israel	437		972
Italy	437 and 850	it	039
Japan	932, 437 and 850		081
Korea	934, 437 and 850		082
Latin America	437 and 850	la	003
Netherlands	437 and 850	nl	031
Norway	865 and 850	no	047
Peoples Republic of China	936, 437 and 850		086
Portugal	860 and 850	po	351
Spain	437 and 850	sp	034
Sweden	437 and 850	sv	046
Swiss (French)	437 and 850	sf	041
Swiss (German)	437 and 850	sg	041
Taiwan	938, 437 and 850		088
United Kingdom	437 and 850	uk	044
United States	437 and 850	us	001

If you want to display the code page information, you can enter CHCP to display the code page information. You can also enter the KEYB command by itself (this command is described in detail later) and DOS displays the code page infor-

mation for the keyboard and screen. Finally, you can use the MODE command to list various types of information about the devices connected to your system. Figure 15-3 shows the beginning of the output this command produces. As the figure shows, this information includes code page information.

While all of this may seem like a lot of work, it is not really the case since these four steps are usually combined with the CONFIG.SYS and AUTOEXEC.BAT files. If you want to change between several languages, you will want to develop batch files to enter these commands for you. Since batch files can automatically make the change for you, once you create these batch files, you do not have to worry about correctly using these commands. Also, by changing the code page, you are making the computer able to display various foreign characters that are not available with the default. Changing the code page is often combined with changing the keyboard country so the special foreign characters that are frequently used are readily available.

■ ***NOTE:*** If you change the code page, you will want to use the same code page every time you work with files created with a specific code page.

Figure 15-3: Mode Command Showing Code Pages Available and Used

```
Status for device LPT2:
----------------------
LPT2: not rerouted

Status for device LPT3:
----------------------
LPT3: not rerouted

Status for device CON:
----------------------
COLUMNS=80
LINES=25

Active code page for device CON is 850
Hardware code pages:
  code page 437
Prepared code pages:
  code page 850

Status for device COM1:
----------------------
RETRY=NONE

C:\>
```

The MODE command displays the current settings for the devices connected to your system.

Keyboard Codes

Another way you can customize the system for using different languages is by customizing the keyboard to match the language you are using. Changing the keyboard to match the language you are using makes special characters of that language quickly available instead of using the ALT key and the numeric keypad to create special keys. The keyboard is set by keyboard codes with the KEYB command. By using the KEYB command, you can change the keyboard code. You can change the keyboard code with CONFIG.SYS file, the AUTOEXEC.BAT file, or by entering a command at the DOS command prompt. When you change the keyboard code, you are changing how DOS interprets each keystroke. For example, when you type a colon (:) with the keyboard set to the United States (the default), the computer interprets the keystroke as a colon. If you change the keyboard code to Latin America, the computer will interpret this keystroke as Ñ. Each country has a different keyboard code. The different keyboard codes are listed in the third column of Table 15-10.

■ ***NOTE:*** Changing the keyboard code affects all software you use once the keyboard code is loaded.

To change the keyboard code in CONFIG.SYS file, include a line like :

```
INSTALL=KEYB.COM  COUNTRY,CODE PAGE,LOCATION OF KEYBOARD.SYS
```

To change the keyboard code in the AUTOEXEC.BAT file or by typing it directly at the DOS prompt, enter:

```
KEYB  COUNTRY,CODE PAGE,LOCATION OF KEYBOARD.SYS
```

For country, you must enter one of the two letter keyboard codes listed in the third column of Table 15-10. For code page, you must enter the code page to use only if you want to change it. For location of KEYBOARD.SYS, this is the path and file information of KEYBOARD.SYS—if it is not in the root directory. The KEYBOARD.SYS file contains the information the computer uses for each country's keyboard. When you enter the KEYB command, DOS sets the exit code to one of the values in Table 15-11.

As a couple of examples, you can change the keyboard to the Latin America keyboard by entering in the AUTOEXEC.BAT file:

```
KEYB LA,850,C:\DOS\KEYBOARD.SYS
```

Table 15-11: KEYB Exit Codes

KEYB Exit Codes	Meaning
0	Program loaded successfully
1	Invalid syntax, keyboard code, or code page
2	Keyboard definition file missing or bad
4	Error occurred with CON device
5	Code page requested not prepared

This example changes the keyboard to the Latin America keyboard which among other changes has the Ñ to the right of the L and has the ¡ and the ¿ above the 1 and 2 (on all keyboards except the enhanced). This example loads and uses code page 850 and tells DOS that it can find the file KEYBOARD.SYS in the C:\DOS directory. Another example using the CONFIG.SYS file is:

```
INSTALL=KEYB.COM GR
```

This example changes the keyboard to the German keyboard. This keyboard has the Ä and Ö to the right of the L. This example assumes that you do not want to change the code page and the KEYBOARD.SYS file is in the root directory.

Once the keyboard is loaded, you can press CTRL-ALT-F1 to return to the default keyboard format, usually the United States. When you want to return to the keyboard selected with the KEYB command, press CTRL-ALT-F2. A third option is to press CTRL-ALT-F7 to switch to the typewriter mode which is standard for some countries.

■ ***NOTE:*** If you change the keyboard code, label keys that change so users can find the keys they want.

Changing DOS Conventions to Match Another Language

The third type of change you may want to make is to change the conventions DOS uses. These changes include the date and time format, the currency symbol, the sort order, and the characters that you can use in file and directory names. These changes are made with the COUNTRY statement in the CONFIG.SYS file.

To change the country, include a statement like this in your CONFIG.SYS file:

```
COUNTRY=country code,,C:\DPS\COUNTRY.SYS
```

The country code number is a three-digit number—these are shown in the last column of Table 15-10. This number is also the international country code for the telephone system. The C:\DOS\COUNTRY.SYS is only required if the COUNTRY.SYS file is not in the root directory. If COUNTRY.SYS is not in C:\DOS, you must provide the correct drive and path information. You can also set the code page by including the code page number between the two commas as in COUNTRY=003,850,C:\DOS\COUNTRY.SYS. An example of the different time and date formats for the country codes are shown in Table 15-12. Changing the sort order causes DOS to sort (as in the SORT command) so the alphabetic order matches the alphabetic order of the country selected.

Table 15-12: Country Customizations

Country/Language	Date Format	Time Format
Arabic	30/10/1991	10:00:00,00
Belgium	30/10/1991	22:00:00,00
Denmark	30-10-1991	22.00.00,00
English (International)	30-10-1991	10:00:00,00
Finland	30.10.1991	22.00.00,00
France	30/10/1991	22:00:00,00
French Canadian	1991-10-30	22:00:00,00
Germany	30.10.1991	22.00.00,00
Israel	30 10 1991	22:00:00.00
Italy	30/10/1991	22:00:00,00
Japan	1991-10-30	22:00:00.00
Korea	1991-10-30	22:00:00.00
Latin America	30/10/1991	22:00:00.00
Netherlands	30-10-1991	22:00:00,00
Norway	30.10.1991	22.00.00,00
Peoples Republic of China	1991-10-30	22:00:00.00
Portugal	30/10/1991	22:00:00,00
Spain	30/10/1991	22:00:00.00
Sweden	1991-10-30	22:00:00.00

Country/Language	Date Format	Time Format
Swiss	30-10-1991	22.00.00.00
Taiwan	10-30-1991	22:00:00.00
United Kingdom	30-10-1991	10:00:00.00
United States	10-30-1991	10:00:00.00

Examples of Changing the International Settings

Since changing the international settings of DOS has several steps, you will want to review these examples so you will be familiar with the commands you must use when you prepare to make your own changes. One example is setting DOS to use the Latin American conventions and keyboard and to use code page 850. The CONFIG.SYS file might include:

```
COUNTRY=003,850,C:\DOS\COUNTRY.SYS
DEVICE=C:\DOS\DISPLAY.SYS  CON=(EGA,437,1)
```

The AUTOEXEC.BAT file in this example would look like this:

```
NLSFUNC
MODE CON CP PREP=((850)C:\DOS\EGA.CPI)
CHCP 850
KEYB LA,,C:\DOS\KEYBOARD.SYS
```

The first line in the CONFIG.SYS files sets the country to Latin America using code page 850 and tells DOS that the COUNTRY.SYS file is in the C:\DOS subdirectory. The second line loads a device driver for the EGA or VGA monitor with the hardware code page of 437 and reserves room for one prepared code page. If you were also setting the code page for your printer, the DEVICE command for the printer would be next. Since it is not included, the printer will use the character sets that are built into the printer.

In the AUTOEXEC.BAT file, the first line loads the NLSFUNC program, the second line loads the prepared code page which is stored in the file C:\DOS\EGA.CPI, the third line changes the code page to 850, and the fourth line sets the keyboard to use the Latin America keyboard. The commands in the AUTOEXEC.BAT file could be stored in a batch file if you only want the international settings for DOS changed on demand, rather than setting it every time you load DOS. If you use a batch file, you might use a second batch file to return DOS to the United States settings. That batch file would look like this:

```
CHCP 437
KEYB US,,C:\DOS\KEYBOARD.SYS
```

After performing the second batch file, the code page would return to 437 and the keyboard would return to the United States default. The international conventions set with the COUNTRY command would still be in effect. This would affect how the date and time appear, how international characters are sorted with the standard English alphabet, and the default currency symbol.

Another example is changing the codepage to 850 without changing the country or keyboard. This would be the case if you wanted to use more international characters than are available on code page 437. In this case, the CONFIG.SYS file would include:

```
DEVICE=C:\DOS\DISPLAY.SYS  CON=(EGA,437,1)
```

The AUTOEXEC.BAT file in this example would include:

```
NLSFUNC
MODE CON CP PREP=((850)C:\DOS\EGA.CPI)
CHCP 850
```

In this example, the DEVICE command in the CONFIG.SYS files reserves space for the prepared code page. The NLSFUNC command loads the program DOS uses to prepare manipulated code pages. The MODE command loads the code page into memory. The CHCP command changes the code page to 850.

At a Glance

In this chapter you have learned how to make many different customization changes to DOS. Some of the main customization options you learned to do in this chapter are:

- **Change the appearance of the DOS prompt—PROMPT**
- **Change how DOS interacts with your system—Create or modify CONFIG.SYS**
- **Execute DOS commands when you boot your system—Create or modify AUTOEXEC.BAT**
- **Stop a DOS command—CTRL-C**

- **Set when DOS checks for CTRL-C—BREAK**
- **Allow DOS to open additional files concurrently—FILES**
- **Add device drivers to CONFIG.SYS—DEVICE or DEVICEHIGH**
- **Change the screen appearance or reassign keys—Load ANSI.SYS**
- **Change the code page for international characters—CHCP**

Performance Tuning DOS

You can increase DOS's performance with a few changes. Since DOS performance affects how well your application software runs you will be able to determine that DOS's performance is better through improvements in your other programs.

You can affect DOS's performance with different types of changes. You can cause DOS to use your disk drives more efficiently. You can improve how quickly DOS sends information to the screen. You can change how DOS uses the memory of your disk and of RAM. With many of these performance enhancements, there are trade-offs. A performance improvement in one area may cause a degradation in another, making it necessary to carefully weigh and balance the changes that you make. Deciding which performance enhancements you will use depends on your hardware and experimentation to determine which changes are providing performance improvements. For example, if you have limited memory in your machine, it may not be a good idea to use a RAM drive. Although the RAM drive will make disk input and output operations faster, you may not have sufficient memory remaining to run application software that you want to work with.

What You'll Learn

Most of the changes made in this chapter are made using the CONFIG.SYS and the AUTOEXEC.BAT files. Chapter 15 showed several commands you may want to include in these files. You may want to review sections of that chapter as you try different performance enhancements. In this chapter you will learn to:

- **Minimize fragmentation**
- **Set the number of buffers DOS uses**
- **Use the SMARTDrive and Fastopen to use files faster**
- **Improve screen speed**
- **Use extended, expanded, and reserved memory and free memory**
- **Set up a virtual disk**

Before You Change Your AUTOEXEC.BAT and CONFIG.SYS Files

Since some of the changes you initiate as you try to make your computer work better may not work, you should create a disk that will allow you to restart the computer if a change does not work out. To do this, format a disk with the command **FORMAT A: /S**. This formats a disk in drive A to include the system files that DOS needs to boot. Next, copy the AUTOEXEC.BAT and CONFIG.SYS files to this diskette. When you run into a problem, you can boot the system using this diskette.

Improving Disk Efficiency

You can improve your system's performance by making your disk more efficient. Disks perform better when they are well organized. In Chapter 8, you learned how you can organize your files using subdirectories. You can also make your disk more efficient by storing all the data for a file in one location on the disk. Other options for making disks more efficient include creating *caches* (temporary storage areas) for files and for storing the filenames and locations of the files you use frequently.

Improving Disk Performance by Organizing Your Disk

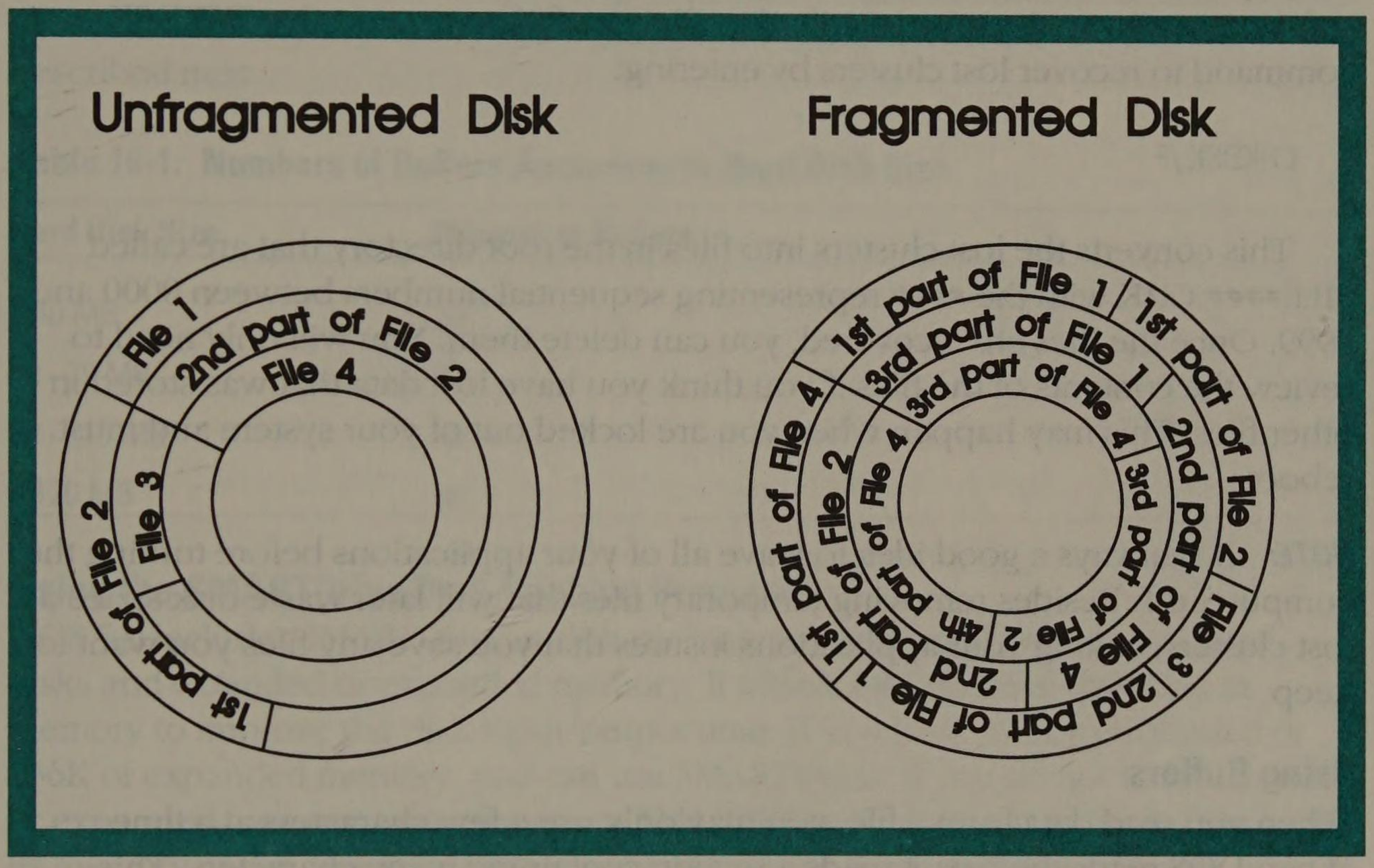
Each time DOS writes a file to a diskette, it uses the first available areas on your disk. As you read and write more information to the disk, the parts of a file may be stored in different locations. This is called *fragmenting*. As a disk becomes more fragmented it means that DOS must spend more time relocating the read/write head to where each part of the file is located. Also, if a disk error occurs, it is harder to recover data from a fragmented disk. You can improve performance with read and write tasks by decreasing fragmentation. Figure 16-1 provides an example of fragmented and unfragmented files. Fragmentation is less of a problem with smaller capacity diskettes since they contain fewer files and are less likely to be split up. You can reduce fragmentation in several ways.

One method is to delete unnecessary files. Deleting unneeded files makes the space they used available for other files. This insures that you have room for other files you need and lets DOS manage the information for only the files you need. Keeping unneeded files is like maintaining a cluttered desk. Every time you need a piece of paper you need to rummage through all the clutter and waste time.

Another option is to backup your hard disk, reformat it, and restore the files to the hard disk. This copies the backed up files one by one to the newly formatted disk so the files are no longer fragmented. Depending on the size of the disk, this

can take time although with tape backup devices, you can backup and restore files with little need for your involvement. This method eliminates all fragmentation and although it takes longer than deleting unneeded files, it is the only way to eliminate all the fragmentation. The BACKUP and RESTORE commands are covered in Chapter 7.

Figure 16-1: Fragmented and Unfragmented Disks


Unfragmented Dlsk
File 1
2nd part of File 2
File 4
File 3
1st part of File 2
Fragmented Dlsk
1st part of File 1
1st part of File 2
3rd part of File 1
2nd part of File 4
3rd part of File 4
3rd part of File 2
4th part of File 4
2nd part of File 1
1st part of File 4
2nd part of File 3

A third option is to purchase a utility like Mace Utilities, which rearranges how files are stored so they are not fragmented. Before you do this, you should not have any applications running. This means that you must exit the DOS Shell using ALT-F4 or the File Exit command rather than by pressing SHIFT-F9 or selecting Command Prompt from the task list. The way these types of utilities run vary with the product so you must review the documentation that accompanies the product for directions. This usually takes less time than the BACKUP and RESTORE commands, yet provides similar results.

■ ***NOTE:*** Fragmentation is a problem as you frequently add and remove files from a disk. If the files you work with do not change, such as when you use the hard disk to store programs and store your data on floppies, fragmentation is less of a

problem. On the other hand, if you are using the computer for word processing and add and remove files frequently, the disk becomes fragmented quicker.

Using CHKDSK

You can recover disk space using the CHKDSK command. The CHKDSK command finds lost file allocation units or clusters which are neither marked as free to store your files or marked as used by other files. This frequently happens when you end a computer session by turning off the computer without exiting the software you are using. As described in Chapter 6, you can use the CHKDSK command to recover lost clusters by entering:

```
CHKDSK /F
```

This converts the lost clusters into files in the root directory that are called FILE####.CHK with the #### representing sequential numbers between 0000 and 9999. Once the files are recovered, you can delete them. You will only need to review the contents of the files if you think you have lost data that was stored in other files. This may happen when you are locked out of your system and must reboot.

■ ***NOTE:*** It is always a good idea to leave all of your applications before turning the computer off. Besides removing temporary files that will later waste disk space as lost clusters, exiting your applications insures that you save any files you want to keep.

Using Buffers

When you read data from a file, you may only use a few characters at a time. When DOS reads the data it reads a section containing many characters. This immediately provides you with the data you want and subsequent information stored in the file. This means that if you want the next information in the file, DOS can read the information from the memory buffer rather than needing to perform slower disk input operations. You can make DOS more efficient by increasing the number of buffers. As described in Chapter 15, you can set the number of buffers according to the size of the hard disk you are using instead of the default based on the RAM disk size. The best size to use is shown in Table 16-1.

Besides adding buffers, you can also use a secondary cache which is a secondary part of memory that DOS uses to store the next sequential part of the file that comes after the part stored in the buffer. When DOS needs part of a file, it first checks the buffers and then it checks the secondary cache. By storing more of a file in memory every time DOS reads a file, it makes using the file faster since DOS

waits fewer times for the head to move to the correct location on the disk with the disk spinning at the proper speed. This works best when you are reading a file sequentially which is often done for databases and program development. The BUFFERS command creates secondary caches by entering a command like this:

```
BUFFERS=50,8
```

This command creates 50 primary buffers and eight secondary caches. You do not want to create secondary caches if you are using SMARTDrive which is described next.

Table 16-1: Numbers of Buffers According to Hard Disk Size

Hard Disk Size	Number of Buffers
< 40 MB	20
40 - 79 MB	30
80 - 119 MB	40
> 120 MB	50

Using the SMARTDrive Disk Caching Program

DOS 5.0 includes SMARTDrive, a disk caching program you can use with hard disks and extended or expanded memory. It allows DOS to store disk data in memory to improve the disk input/output time. If you have 512K of extended or 256K of expanded memory, you can use SMARTDrive. If you do not have extended or expanded memory, you cannot use SMARTDrive. This program reads files into its memory cache. Since SMARTDrive uses extended or expanded memory, you can let it read larger files than the BUFFERS command uses. When you modify or create a file, SMARTDrive immediately saves the changes so you will not lose them.

To use SMARTDrive, you need to add a line to your CONFIG.SYS file that looks like this:

```
DEVICE=C:\DOS\SMARTDRV.SYS  1024
```

This command assumes that the SMARTDRV.SYS file is in C:\DOS. The number after SMARTDRV.SYS is the size of the disk cache in kilobytes. This example creates a disk cache of 1 MB. The number should be smaller than the total expanded or extended memory if you want to use some of the type of memory for other applications. If you do not provide the cache size, DOS uses 256K. If the

number is too large, DOS creates as large a disk cache (up to 2 MB) as possible. Your system will work best when this number is as large as 2,047 while the computer is able to run your other applications. If you want to provide the minimum size of the cache, you will need to have this number after the size. This prevents other applications from reducing the size of the cache to zero. Since DOS will by default install SMARTDrive in extended memory (unless it is told not to), you must use the /a switch at the end if you want the program installed in expanded memory. Like other commands that use extended or expanded memory, this DEVICE command must come after the command that loads the extended or expanded memory manager. You can use DEVICEHIGH instead to load the SMARTDrive program into reserved memory instead of conventional memory. The disk caching will perform better in extended rather than expanded memory in 80286 or 80386 machines. But you will want to decide which type of memory to use based on how much you have of each memory and what you are using the other memory for with other DOS commands and with your applications.

- ***NOTE:*** SMARTDrive's performance decreases as fragmentation increases. If you are using SMARTDrive and you add and remove files to the hard disk frequently, you will want to compact the hard disk in order to decrease fragmentation.

Using Fastopen to Use Files Quickly

Another program which can improve performance is the Fastopen program. It lets you access files quicker by remembering the files and directories you use. When you use a particular file or directory again, the Fastopen program directs DOS to where it can immediately find the file—this is quicker than following all of the steps to find a file or directory. Fastopen provides the greatest performance increase when you are using applications that open and close a lot of files.

To use Fastopen, you can include it in your CONFIG.SYS file, run it from a batch program such as AUTOEXEC.BAT, or enter it directly into the command prompt. To use Fastopen in the CONFIG.SYS file, you might include a line like this:

```
INSTALL=C:\DOS\FASTOPEN.EXE C:=32 D:=28 /X
```

This example assumes that the file FASTOPEN.EXE is in C:\DOS. This example tracks the last 32 files you are using on drive C and 28 files on drive D. The number after the equal sign is usually best set to 1 for every megabyte on the hard drive, although it can range between 10 and 999. You can track files on up to 24 hard disks at once although the example only does it for two. The /X at the end tells DOS to use expanded memory. This command cannot use extended memory.

Loading Fastopen during CONFIG.SYS is the only way you can load Fastopen into extended memory. Before you have the INSTALL command for Fastopen, you must have any DEVICE command to run the software that manages the expanded memory. If you do not load Fastopen in the CONFIG.SYS file, you will use conventional memory. To run Fastopen from a batch file or command prompt, you would enter a line like this:

```
FASTOPEN C:=32 D:=28
```

Once you start Fastopen, the settings remain in effect until you restart the computer.

Improving Screen Speed

The display that you see on your screen can be graphics or text as described in Chapter 4. A graphics display provides the nicest images since you can put any information in any location on the screen. With text, you are limited to putting characters only in predefined columns and rows. Many applications, including the screen when you see the DOS command prompt, use a text screen. Text screens are faster. With some programs, you can select whether you are using a text or graphics screen. If you have this choice, you will want to use a text screen to make the program run faster. On the other hand, you may want to use a graphics screen when you are using the graphical features of the program. Two examples of a text screen versus a graphics screen are the DOS Shell program and 1-2-3 Release 2 when used with Allways or Impress. With the DOS Shell program, you can run it in text mode by starting it with the /T switch or in graphics mode by starting it with the /G switch. You can also select the mode with the Display Options command in the Options menu.

When the DOS Shell uses text mode, DOS does not draw the icons that represent different drive types, directory folders, or programs and program groups. On the other hand, when you change the information that is displayed on the screen, a text display performs the update faster than in graphics mode. With a program like 1-2-3, when you are using an add-in like Allways or Impress, you use a text mode for working with the spreadsheet data and a graphics mode for customizing the appearance of the data such as by adding fonts, boldface, and other attributes. In this case, you are selecting the screen mode by whether the add-in is invoked.

Improving Memory Utilization

One of the resources your computer has to run applications is memory. A computer can have different kinds of memory as part of the RAM (random access

memory) that it uses to run your programs and the DOS operating system. You can also display the memory that you have available so you can plan how you want to use it. Also, if you have extended memory on a 80386 or 80486 computer, you can use some of the extended memory as expanded memory which other programs and DOS commands like FASTOPEN can use.

Different Types of Memory

Your computer can have four types of memory that you can use. The amount of memory available for each type depends on the hardware in your computer. By knowing the type of memory you have, you can select how DOS and your applications will use this memory.

The four types of memory are conventional, reserved, expanded, and extended.

Conventional is the RAM that all machines have. This can be up to 640K which is stored on the main board (called the mother board). DOS and your applications use at least some of this type of memory.

Reserved memory is available on AT class and newer machines. This is usually 384K and is on the mother board. DOS normally uses this area for device drivers. Most programs do not use this type of memory—older machines do not even have reserved memory.

Expanded memory is memory stored on a separate card in the machine that provides additional memory. This type of memory can be used by all DOS-based machines. This type of memory requires that the programs and applications you use be designed to use expanded memory in order to use this additional memory. For example, spreadsheet programs will use expanded memory to store larger spreadsheets. Not all machines have expanded memory. If your machine has expanded memory, your CONFIG.SYS file will need a DEVICE command to load the software that accesses the expanded software before you can use it.

Extended memory is memory that is part of the RAM that extends beyond 1 MB. You can also use this memory for DOS and applications assuming the DOS commands or applications are designed to use extended memory. Since extended memory is newer than expanded memory, less programs are designed to take advantage of it. This type of memory is available on the AT class and above machines. If you have a 386 or 486 machine and lots of extended memory, you can use part of it as expanded memory.

Displaying Available Memory

You have several choices to display the memory you have available. One of the easiest ways of displaying the memory that you have is when you boot the computer. When you boot the computer, the computer checks that the memory in

the computer is working. This count which you see in the upper left corner of the screen displays the memory that it has checked (this feature also lets you know if you have a problem with the machine). The memory that is measured is conventional and extended memory. If the number is greater than 640K, you have extended memory. If you have expanded memory, the software that is run to use this memory will usually describe the size of the expanded memory. This is also the case when you use part of the extended memory as expanded memory.

Another method of displaying the memory you have available is with the MEM command. As described in Chapter 15, you can type **MEM** and press ENTER to list the conventional, reserved expanded, and extended memory you have available. You can also use the /program or /debug switch for this command to list the starting point and sizes of the programs and applications loaded into memory.

Using Extended Memory

If you have extended memory, you can make it available to DOS and other applications by loading the extended memory manager utility called HIMEM. To load this utility, include a line like this in your CONFIG.SYS file:

```
DEVICE=C:\DOS\HIMEM.SYS
```

This assumes the HIMEM.SYS file is in C:\DOS. Once this utility is loaded, you can load and run other applications that use the extended memory. This includes the DOS command described in Chapter 15 and the EMM386 command described below. Loading this utility also lets you use reserved memory for other tasks.

Using Extended Memory for Your Applications Once the extended memory manager is loaded, you can use it for your applications. Some programs are created to use extended memory if it exists. Often you do not need to do anything to tell it to use the extended memory. Some programs, like Quattro Pro, require you to tell it that extended memory is available by providing information during program setup or execution. For example, with Quattro Pro, you tell it that it has extended memory by using the /X switch after the Q to start executing the program. You can also use extended memory for the DOS SMARTDrive and RAMDRIVE commands. Commands like SMARTDrive automatically use extended memory. Commands like RAMDRIVE only use extended memory if you tell it to.

Using Reserved Memory

If you have reserved memory, you can use it to store device drivers and programs. To make reserved memory available, you must use the DEVICE=C:\DOS\EMM386.EXE and DOS=UMB, HIGH commands in your CONFIG.SYS file. At the end of the

DEVICE command, add RAM if you are using expanded memory or no EMS if you are not.

Not all programs will work in reserved memory so you should try loading a program into reserved memory, checking that it works, and determining that the program is actually using reserved memory rather than conventional memory. First, load the device driver or program in conventional memory (using the DEVICE command or in the way you normally load the program). To use ANSI.SYS as an example of a device driver, include DEVICE=C:\DOS\ANSI.SYS in the CONFIG.SYS file when you boot the computer. To use the PRINT command as an example of a TSR program, type PRINT and press ENTER twice, the second time to use the default printer output. Then type **MEM /PROGRAM >PRN:** and press ENTER. Notice where the program or device driver is on the list and circle it (programs may have more than one line). Remove the device driver from the CONFIG.SYS file and reboot the machine or remove the program from the machine by rebooting the machine. To load a device into reserved memory, use DEVICEHIGH as in:

```
DEVICEHIGH=C:\DOS\ANSI.SYS
```

To load a program into reserved memory, load the program by using LOADHIGH (or its abbreviation of LH) as in:

```
LOADHIGH PRINT
```

For the device driver or program, DOS first checks if reserved memory has enough room for the device driver or program. If it does, DOS loads it into reserved memory; if not, DOS loads it into conventional memory. Next, check that the device driver or program works as planned. With a device driver like ANSI.SYS DOS will display a message if it cannot load the device driver. For a program like PRINT, you can test that PRINT functions correctly by printing a text file. Finally, check that the device driver or program is loaded into reserved memory rather than conventional memory by typing **MEM /PROGRAM >PRN:** and pressing ENTER. Compare the lines for the device driver or program as it is listed in the first printout with the second. You can tell if the device driver or program is loaded into reserved memory if it is listed further down in the output and the address in the first column is greater than C8000. DOS device drivers that work in reserved memory include EGA.SYS, ANSI.SYS, DISPLAY.SYS, SMARTDRV.SYS, DRIVER.SYS, PRINTER.SYS, and RAMDRIVE.SYS. The DOS commands which you

can load and use in reserved memory include DOSKEY, DOSSHELL, APPEND, GRAPHICS, KEYB, MODE, NLSFUNC, PRINT, and SHARE.

- ***NOTE:*** If you are loading a device driver that expands as you use it, you must reserve space for the device driver by including SIZE in the DEVICE command, as described in Chapter 15.

Using Extended Memory as Expanded Memory

If you have lots of extended memory and you are using programs that can use expanded memory but cannot use extended memory, you may want to treat part of the extended memory as expanded memory. If you have a 386 or 486 machine, DOS includes the software to do this. The 386 or 486 refers to the 80386 or 80486 CPU that is inside the machine. You can find out the type of machine you have by looking at the front of the machine or reviewing the documentation that accompanied the machine. Some software products such as 1-2-3 with its /Worksheet Status command and WordPerfect with its WPINFO program will display the processor. Other potential numbers include 8086, 8088, and 80286. To use some of the extended memory for expanded memory, enter these two lines in the CONFIG.SYS file:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE  2048
```

The first line loads the extended (or high) memory manager. Once the computer knows how to use the extended memory, the second line allocates 2048K or 2 MB of extended memory for expanded memory. This number can be as much of the extended memory as you want or you can omit this number and use the default of 1 MB or 1024K.

Freeing Up Memory

One of the most limited resources of a computer is the conventional RAM. This is because most programs are designed to use it for the program and for the data you work with. If your system does not have enough available RAM to run a program, you cannot run it without removing other programs from memory or by adding more RAM. If you need more memory for the programs you are running or the data files you want to use, you can use several options.

Limiting DOS Commands in Memory

Some commands like the PRINT command remain in memory after you use it. You can retain more RAM for your other applications by reducing the DOS

commands that are in memory. These commands and approximations of the memory they use are listed in Table 16-2. While you may provide more conventional memory to your system by limiting the number of DOS commands you use, remember that the features which the commands provide will still not be available for your use. For example, if you are printing graphics screens using SHIFT-PrtSc, you must use the GRAPHICS command even though the command uses 5,840 bytes of conventional memory. For some commands there are other solutions. For example, rather than using the PRINT command to print a file, you can use the COPY command using LPT1 as the destination. Another solution may be to load them in reserved memory as described previously.

Table 16-2: DOS Commands that Remain in Memory

DOS Command	Memory Used
BUFFERS	512 bytes / buffer
FASTOPEN	3K + 48 bytes / file
SMARTDRIVE	21K
HIMEM	1K
EMM386	7K
RAMDRIVE	1K
PRINT	5264 + bytes
GRAPHICS	5840
ANSI.SYS	4480
GRAFTABL	1200
SHARE	6336
DOSKEY	4048

Reviewing Your CONFIG.SYS and AUTOEXEC.BAT Files

You can also free up memory by only including the commands that you need in the CONFIG.SYS file and AUTOEXEC.BAT files. REM commands in either file are ignored by DOS, so you do not have to make a special point of removing them. In the CONFIG.SYS files, you will want to look for DEVICE commands that load device drivers you do not use. For example, if you do not have a mouse, you would not use the DEVICE=MOUSE.SYS command. In the AUTOEXEC.BAT file, you would look to see if it loads any programs you do not need. This includes TSRs (described below). If you are short of memory, you will want to review the

contents of both files to check that only the needed commands are included. If either file contains commands that you may want later, put REM before them so they are ignored by DOS.

Selecting TSRs To Load

TSRs are programs that remain in memory so their features are available when you need them. TSRs provide a wide range of features that are only limited by the programmer's intentions and ability. For example, you may have a TSR like Metro that pops up on the screen to allow you to record notes while you are working with other software. You can purchase all sorts of TSRs as well as use DOS commands that behave like TSRs. For example, the PRINT command remains in memory to print files in the background. If you need conventional memory for other applications, you may not want to load all of your TSRs or you may want to try loading them into reserved memory. These may be loaded when the AUTOEXEC.BAT file is executed, or directly from the DOS command prompt. You may want to remove the line in the AUTOEXEC.BAT file or not start the TSR from the DOS prompt. Another option is to unload TSRs when you are finished using them. TSRs are unloaded when you restart the computer, although many of them are also unloaded with a command or special key sequence. When you unload TSRs, you must unload them in last in, first out (LIFO) order. This means that if you first load the Metro TSR and then load WordPerfect's Grab TSR (it saves graphic images that you can later incorporate into WordPerfect's documents), you must unload Grab before you can unload Metro.

- ***NOTE:*** When you are using a TSR in the middle of an application, you will want the screen to be in text mode. Most TSRs use text mode, so trying to put a text mode TSR on top of a graphics mode screen results in something that is illegible. A few TSRs will adjust themselves to the graphical mode in effect when they are invoked, but you can find this out by reading the instructions that accompany the TSR.

Setting Up a Virtual Disk

Another use of memory is to use a virtual disk. For example, if your hard disk is almost full and you are working with data on diskettes, you may want to create a virtual disk to temporarily store the information on the diskettes so the application that uses the data files is faster. This is because reading and writing to a floppy disk drive is one of the slowest operations a computer can perform. You can section off part of the computer's memory to be treated as if it is a disk. A disk created by temporarily using the computer's memory is called a *virtual disk*. Like

other parts of the computer's memory, when you turn off the computer or reboot the system, the information stored in the memory as well as the virtual disk is erased. While virtual disks are fast, you must always make sure that you copy the files to more permanent media when you are finished. To create a virtual drive, you would enter a command like this in your CONFIG.SYS file:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 360 512 112 /e
```

You must use a command like this for each virtual drive you want to create. This example assumes that RAMDRIVE.SYS is in C:\DOS. The first number after RAMDRIVE.SYS is the number of kilobytes the drive should contain. You can omit this number and the next two, if you want to use the default of 64K. The second number, if it is used, is the number 128, 256, or 512—representing the number of bytes. You can also specify a third number which is the number of entries (minus one) that you will be able to create in the root directory. This number sets the size of the area DOS reserves for the root directory which can be a number between 2 and 1,024. If you do not provide this number, DOS uses 64. At the end you can use /e to use extended memory for the virtual disk, or /a to use expanded memory. If you do not use either, DOS uses conventional memory. If you are creating a small virtual drive, you can use DEVICEHIGH to create a virtual drive in reserved memory. Since this greatly reduces the memory other applications use, you will usually create virtual drives if you have extended or expanded memory. The DEVICE command that creates the virtual drive in extended or expanded memory must come after the command that loads the extended or expanded memory manager if it uses extended or expanded memory. If you want multiple virtual drives, use multiple DEVICE=C:\DOS\RAMDRIVE.SYS commands in the CONFIG.SYS file.

Once you create a virtual disk, you can use it to store any type of information. The virtual disk will be assigned the next unused logical drive letter. If you are using the DOS Shell with a graphical display, the virtual disks will display RAM to indicate that they are created with RAM memory. With the virtual disks, you can use any command except FORMAT and DOS will treat the virtual drive as it treats other hard disks. You will notice that the virtual drive is fast. When you are finished using the files on the virtual disk, make sure to copy the files you want to another disk. Once the computer is turned off or rebooted, there is no way you can get the data you previously had stored on it.

- ***NOTE:*** If you get the message that DOS cannot make a directory entry onto the hard drive, it is because the root directory is full with entries. You can either create a subdirectory (after deleting a file since a directory takes one slot in the root

directory) or you can create your virtual disk using the three numbers after RAMDRIVE.SYS, and providing a higher number. The default is only 64, but on large virtual disks, this number will be insufficient. Subdirectories on virtual disks do not have any limits on the number of files.

- ***NOTE:*** Some programs create temporary files and use the environment variable TEMP to select the location of these temporary files. If the program does not find a TEMP environment variable, it uses the current directory. You can speed up reading and writing temporary files by entering **SET TEMP=E:** assuming E:\ is your virtual drive.

Using Windows

Windows is a program developed by Microsoft that lets you work with several programs simultaneously. With Windows, you use the optimization procedures described in the *Microsoft Windows User's Guide* rather than the optimization procedures described in this chapter. If you use Windows 3.0 or later, you do not use the EMM386 command to treat some extended memory as expanded memory since Windows takes care of it for you. You should not use SMARTDrive with other disk caching programs or version 3.0 with Windows 2.x. SMARTDrive works better than other disk caching programs with Windows 3.X since they are designed to work together.

Additional DOS Enhancements

DOS has other features you can use to enhance your productivity with DOS. The following suggestions can be used to change how you use programs and how DOS responds. You can make copying files more efficient by using the XCOPY command instead of the COPY command since XCOPY reads and copies multiple files at a time instead of one at a time. When you organize a hard disk, keep subdirectories small so both you and DOS will find files faster. You can use the PATH command to tell DOS where it can find the programs you want to run. For example, if you are using 1-2-3, WordStar and Harvard Graphics, your PATH command may be:

```
PATH C:\;C:\DOS;C:\123;C:\WS5;C:\HG
```

If you have extended memory or expanded memory, use it even if your programs don't. You can use this memory for the FASTOPEN command, with SMARTDrive, or for a virtual disk. You can also put DOS and the buffers created

with the BUFFERS command in the high memory area by including a DOS=HIGH command in CONFIG.SYS.

If you have ANSI.SYS loaded, you can make switching between programs easy by assigning DOS commands to keystrokes. The following batch file assigns function keys F5 through F8 to change to different programs:

```
@echo ^[[0;63;"c:";13;"cd\wp51";13;"wp";13p
@echo ^[[0;64;"d:";13;"cd\qpro";13;"q /x";13p
@echo ^[[0;65;"c:";13;"cd\hg";13;"hg";13p
@echo ^[[0;66;"d:";13;"cd\m2";13;"mgfast";13;"d:";13p
```

The first line sets F5 to change the drive and directory to C:\WP51 and start WordPerfect. The second line sets F6 to change the drive and directory to D:\QPRO and start using Quattro Pro with extended memory. The third line sets F7 to change the drive and directory to C:\HG and start Harvard Graphics. The fourth line sets F8 to change the drive and directory to D:\M2 and start Magellan. Chapter 15 includes the ANSI.SYS codes you use to assign keys. Once you execute this batch file, you can press one of the four function keys to start a program from the command prompt. While you are in the program the assigned keys will use their temporary key assignments that the program uses, and then return to the assignments created with the batch file when you return to DOS.

You can also make your keyboard respond faster, especially when you repeat keys, such as pressing ENTER multiple times. This is useful for moving around in a document in your word processor as well as playing a computer game when you are moving from one position to another. You can make the keyboard react faster to your keystrokes by entering a command like:

```
MODE CON RATE=32 DELAY=1
```

The number after rate is between 1 and 32 which selects how often DOS repeats a key per second when you hold it down. The higher the number, the more frequently DOS repeats the key. The number after the delay is 1, 2, 3, or 4, which represents the number of quarter seconds DOS waits until it recognizes you are intentionally holding down the key.

Chapter 5 described how you can enable the switcher in the DOS Shell to run multiple programs at once. This lets you work with different programs at once and allows you to switch between them to use the programs you want. You can use this feature to make your time with the computer more productive since you spend less time switching between the data you want.

At a Glance

When you are deciding which of the options covered in the chapter you want to use, you must decide how you want to allocate your resources. For example, if you have lots of extended memory, you may want to decide how much you want to use as expanded memory, as a virtual disk or as a disk cache with SMARTDrive. Some of the options you can try for enhancing DOS performance are:

- **Reduce disk fragmentation—Delete unneeded files, use BACKUP and RESTORE, or use a separate utility**
- **Recover lost disk clusters—CHKDSK /f**
- **Optimize the numbers of buffers—BUFFERS=**
- **Use SMARTDrive for disk caching—Add DEVICE=C:\DOS\SMARTDRV.SYS to CONFIG.SYS**
- **Use FASTOPEN for quick file access—Add INSTALL=C:\DOS\FASTOPEN.EXE C:=32 D:=28 /x to CONFIG.SYS**
- **Display available memory—MEM**
- **Use extended memory—Add DEVICE=C:\DOS\HIMEM.SYS to CONFIG.SYS**
- **Use extended memory for expanded memory—Add to CONFIG.SYS DEVICE=C:\DOS\HIMEM.SYS and DEVICE=C:\DOS\EMM386.EXE 2048**
- **Create a virtual drive—Add DEVICE=C:IDOS\RAMDRIVE.SYS 360 512 112 /e**

Installing MS-DOS 5.0

Installing DOS 5.0 is easier than installing earlier versions of DOS. The process has been streamlined, and the program is able to determine the best course of action without your intervention, in most cases. DOS looks at your configuration for earlier releases if you are upgrading and attempts to duplicate the same setup. The essence of what you need to do will be indicated through screen messages that are explicit in their directions.

The process of installing MS-DOS 5.0 will differ slightly depending on whether you are installing DOS to a new system without a copy of DOS, or installing it on a system that already has another copy of DOS on it. If DOS is already on your system, you are actually performing an upgrade procedure rather than a new installation. Installation and upgrade procedures will be discussed in separate sections since the process and requirements are a little different.

DOS 5.0 can be used for systems with a hard disk as well as those that are limited to floppy disks. Since the information on your DOS disks is stored in a compressed format, you will need blank disks for storing the information that the DOS disks contain in an uncompressed format.

DOS 5.0 Requirements

Your computer system must meet certain minimum requirements if you plan to successfully install DOS 5.0 on it.

First, it must be based on a chip architecture for the Intel 8086 processor or higher. This means that you can run DOS 5.0 on any machine from an 8086 to a 286 or 386 machine. These machines include IBM PC's, IBM XT's, IBM AT's, IBM PS/2's and any machines that are compatible with these models.

Second, your machine must have at least 256K of memory for a floppy disk system, and 512K of memory for a hard disk system.

Third, your system must have at least 2.5 MB of hard disk space for a hard disk system. You must install DOS 5.0 on the primary partition of your first hard disk drive, and this drive must be active.

Upgrading Your System To MS-DOS 5.0

You can upgrade your system from any version of DOS from 2.11 or later to DOS 5.0. When you already have a copy of DOS on your system, DOS will make some assumptions about how you want the new release to be configured and ask you to verify its selections. Since you have a working copy of DOS on your system, you have probably created files for various applications that you will want to backup and the upgrade procedure offers you this option. Upgrade also gives you the capability to restore your previous release of DOS.

With a floppy disk system you will begin using your new DOS disks after the installation. You will be able to return to your earlier DOS release (2.11 or later) by placing it in your drive before booting your system. With a hard disk system there can only be one current release of DOS in the root directory, and the upgrade procedure will take a few additional actions:

- Copies your old AUTOEXEC.BAT and CONFIG.SYS files and gives the copies a number as a filename extension.
- Modifies your AUTOEXEC.BAT and CONFIG.SYS files for DOS 5.
- Creates a directory called OLD_DOS and gives it a number as an extension. DOS copies all of your old DOS files to this directory.
- Creates an Uninstall disk containing information about your previous DOS release including a copy of AUTOEXEC.BAT and CONFIG.SYS named CONFIG.DAT and AUTOEXEC.DAT. This disk can be used to return to the earlier release of DOS or to boot DOS 5.0 if your hard disk fails.

Copying the Upgrade Disks

The first step in your upgrade procedure is to copy each of the disks in your DOS package. You will want to use DISKCOPY to make the copies; otherwise, you will need to use the LABEL command to make the disk labels on the copies match the labels on the original disks. Follow these steps:

1. Locate the same number of disks that were supplied with your DOS package. Choose disk of the same size and capacity as the original disks.
2. Type **DISKCOPY A: A:** if you have only one floppy drive or **DISKCOPY A: B:** if you have two floppy drives.
3. Follow the DOS prompts for inserting the correct disks. If you have only one drive, each disk will need to be inserted several times in response to the prompts.

4. Respond with Y when prompted about making another copy and proceed with step 3 for the next disk. When all the disks have been copied you can type an N.

Upgrading to a Hard Disk

Once you have copied the disks you are ready to use the setup program to upgrade your system. You will need one high density floppy disks or two low density floppies to create uninstall disks as part of the upgrade procedure. These disks will be used to allow you to return to your previous DOS version and to start DOS from a floppy disk. Follow these steps:

1. Boot your system with the current version of DOS.
2. Place Disk 1 from DOS 5.0 in drive A.
3. Type **a:setup** and press ENTER.

Follow the directions on your screen to complete the upgrade.

Upgrading to a Floppy Disk

When you upgrade a floppy disk system there is no need to create Uninstall disks since you can always return to the previous release by using your old disks. You will need to create a set of DOS disks for your use since many of the files on the original DOS disks are stored in a compressed format. If you are using 5 1/4 inch disks you will need seven disks. You should label them as follows: Startup, Support, Shell, Help, Basic/Edit, Utility, Supplemental. If you are using 3 1/2 inch disks you will only need four disks. You should label them Startup/Support, Shell/Help, Basic/Edit/Utility, and Supplemental. Once your blank disks are labeled follow these steps:

1. Boot your system with your current DOS disk.
2. Place Disk 1 from DOS 5.0 in drive A.
3. Type **a:setup /f** and press ENTER.

Follow the directions on your screen and insert each of the blank labeled disks as requested.

Restoring Your Previous DOS Release

When you perform the upgrade procedure on a hard disk system, DOS creates Uninstall disk(s). The purpose of these disks is to allow you to start DOS 5.0 from these disks if you have a problem with your hard disk, or to restore your previous

version of DOS. Using Uninstall to restore your previous DOS release will not work if you have changed the partitions or formatting of your hard disk, or if you have moved DOS 5.0 files to another location. If you upgraded a floppy disk system all you need to do to use your previous DOS release is to boot your system with the disk that you used before installing DOS 5.0. To use Uninstall follow these steps:

1. Place the disk that you labeled Uninstall #1 in drive A and boot your system.
2. Press F3 when the Uninstall application displays on the screen to work with DOS 5. Follow the directions on your screen to return to the previous DOS release.

Installing MS-DOS 5.0

The installation procedure is almost identical to the upgrade procedure except that it is for users who do not have a prior release of DOS stored on their machine. You will need to book your system using DISK 1 of your MS DOS 5.0 disks. Just as with the upgrade procedure, you can type either **setup** or **setup /f** depending on whether or not you are using a floppy disk system. You will need to follow the instructions on your screen to complete install. Remember, DOS 5.0 files are stored in compressed format. Your DOS 5.0 files will be unusable unless they are installed with setup. If you copy them to your hard disk without setup they will be unusable.

Changing Installation Settings for Your Monitor

A display type is selected during installation to match your current system configuration. If you later change monitors, you will need to reinstall the files for your monitor.

DOS supports the following display types: MONO, CGA, EGA, EGA MONO, VGA, VGA MONO, Hercules, Olivetti, Olivetti MONO. The files needed for each monitor type are listed in Table A-1. You can locate the disks containing the files you need by editing the file PACKING.LST that is provided with DOS 5.0. Follow these steps to change your video files:

1. Insert the disk containing the .VI file you need for your monitor type into drive A.
2. Type **EXPAND A:\NAME.VI_ C:\DOS\DOSSHELL.VID** where name.vi_ is replaced by the name of the appropriate file from Table A-1.
3. Insert the disk containing the .GR file that you need into drive A.
4. Type **EXPAND A:\NAME.GR_ C:\DOS\DOSSHELL.GRB** where name.gr_ is the name of the file from Table A-1 that you need.

5. Copy dosshell.in to dosshell.in2 by typing **COPY C:\DOS\DOSSHELL.IN C:\DOS\DOSSHELL.IN2**.

This step allows you to retain the existing file in the event that you want to return to it. You can also use the COMPARE command on the old and new file to locate the entries for color= and selection = to quickly identify the custom changes that you made to the original file.

6. Insert the disk containing the .IN file that you need for your monitor type in drive A.
7. Type **EXPAND A:\NAME.IN_ C:\DOS\DOSSHELL.IN** where name.in_ is the name of the file from Table A-1 that you need.

Table A-1: Display Types

Display Type	Video Display Files
Monochrome	No .VI_ file, MONO.IN_, MONO.GR_
CGA	CGA.VI_, CGA.IN_, CGA.GR_
MCGA	CGA.VI_, CGA.IN_, VGA.GR_
EGA	EGA.VI_, EGA.IN_, EGA.GR_
EGA Monochrome	EGA.VI_, EGA.IN_, EGAMONO.GR_
VGA	VGA.VI_, EGA.IN_, VGA.GR_
VGA Monochrome	VGA.VI_, EGA.IN_, VGAMONO.GR_
Hercules	HERC.VI_, MONO.IN_, HERC.GR_
Olivetti or AT&T	CGA.VI_, CGA.IN_, OLIV.GR_
Olivetti or AT&T Monochrome	CGA.VI_, CGA.IN_, SV400A.GR_

DOS Error Messages

DOS displays utility messages from the DOS utilities, device error messages, and application program messages. This appendix focuses on DOS utility and device error messages. Many of the informational type messages displayed by DOS are not included since they are designed to provide information on the task DOS is currently performing for you. The messages selected for inclusion include those that might require you to take some action to deal with a problem situation, or a message that informs you of limitations in the current command or utility.

---

10 Mismatches - ending compare

What this means DOS ends the compare process assuming the two files are different when it encounters the tenth mismatch.

How to solve the problem Check to ensure you are comparing the correct files. You may need to make another copy, and compare again.

Access Denied

What this means You are trying to access a file where access is not permitted. This can be caused by accessing a read-only or write protected file, or by trying to look at the contents of a directory file with TYPE.

How to solve the problem Use ATTRIB to remove the read-only attribute. Check to insure that you are not attempting to treat a directory file as a regular file.

All available space in the Extended DOS Partition is assigned to logical drives

What this means All of the space in the extended DOS partition has already been assigned to logical drives and no more space is available to make another assignment.

How to solve the problem Backup the data in one of the current logical drives, delete this drive, and add it and the new drive to the partition, splitting the available space between them.

Allocation error, size adjusted

What this means There is an inconsistency between the amount of data in a file and the amount of data indicated in the directory. DOS will truncate the file to match its allocation.

How to solve the problem Use CHKDSK with the /f switch to correct this error.

ANSI.SYS must be installed to perform requested function

What this means You are trying to perform a screen function that requires ANSI.SYS.

How to solve the problem Use the DEVICE command to load ANSI.SYS. See chapter 15 for more information.

APPEND already installed

What this means You have attempted to use APPEND with a switch that is only valid the first time you use APPEND in a DOS session.

How to solve the problem Reboot your system and enter APPEND with the desired switch. Note—if this message appears again when you try, APPEND is included in your AUTOEXEC.BAT file and you will need to make the modification to this file before rebooting again.

APPEND/ASSIGN Conflict

What this means You are trying to use APPEND on an assigned drive.

How to solve the problem Type ASSIGN with any parameters to cancel the assignment, and try APPEND again.

/B invalid with a black and white printer

What this means You tried to use the /B switch with the GRAPHICS command to assign the background color for printing, yet you have a black and white printer without color capabilities.

How to solve the problem Reexecute the GRAPHICS command without /B.

Bad command or file name

What this means The entry you typed was not recognized by the computer as either a filename or a command to execute.

How to solve the problem Make sure you have spelled the command or filename correctly. If you have, it is not a command; or if it is a file, the file is not located in that sub-directory. Move to the correct sub-directory and re-execute the command. Alternatively, you can enter a PATH statement to tell DOS which other directories it should be checking for programs. You can use APPEND to have DOS check other directories for files.

Bad or missing *d:path* SMARTDRV.SYS

What this means DOS cannot find the file SMARTDRV.SYS.

How to solve the problem Change the PATH statement in CONFIG.SYS to include the directory where SMARTDRV.SYS is located.

Bad or missing Command Interpreter

What this means DOS is either unable to locate COMMAND.COM or the copy it has located is bad.

How to solve the problem Restart your system with a disk that contains COMMAND.COM. If you are using a hard disk you will then have to copy COMMAND.COM from your backup DOS disk to the hard drive.

Bad or missing *filename*

What this means Your CONFIG.SYS file contains a DEVICE statement with a misspelled filename, or a location for the filename that is incorrect.

How to solve the problem Change CONFIG.SYS to fix the incorrect DEVICE statement.

Bad or missing keyboard definition file

What this means DOS is unable to locate KEYBOARD.SYS, or the file is corrupted.

How to solve the problem Check the locations specified in the PATH statement to see if KEYBOARD.SYS exists. If you find it you will need to replace it with your

backup DOS disk. Use ATTRIB /S to see if the file is located in a directory not included in the PATH statement, and revise the PATH statement if necessary.

Bad Partition Table

What this means DOS is unable to locate a partition on your disk.

How to solve the problem Run FDISK to create a partition and the required logical drives.

Bad unit error reading drive *x:*

What this means Invalid information was passed to the device driver.

How to solve the problem Check the syntax of the DEVICE command in CONFIG.SYS.

Batch file missing

What this means After part of the batch file has been executed, the computer cannot locate the remainder of the batch file to execute.

How to solve the problem Re-Insert the diskette containing the batch file.

Baud rate required

What this means A MODE COM command has been specified without the required baud rate.

How to solve the problem Revise the MODE command to include the required information.

Cannot CHDIR to path - tree past this point not processed

What this means CHKDSK is unable to look at the remaining subdirectories.

How to solve the problem Use CHKDSK with the /f option to correct the problem.

Cannot CHDIR to root

What this means CHKDSK is unable to return to the root directory of the disk it is processing.

How to solve the problem Try CHKDSK again. Repeated errors indicate the disk is unusable.

Cannot Chkdsk a Network drive

What this means You have attempted to run CHKDSK on a network drive.

Cannot Chkdsk a SUBSTed or Assigned drive

What this means CHKDSK will not work with drives that have been used with ASSIGN or SUBST.

Cannot create a zero size partition

What this means You have attempted to create a hard drive with a partition size of zero.

How to solve the problem You must allocate at least 1MB or 1% of the disk to a partition.

Cannot create extended DOS partition without primary DOS partition on disk 1

What this means You have attempted to create an extended DOS partition with FDISK, but you have not created the primary partition yet.

How to solve the problem Use FDISK to create the primary partition first.

Cannot create Logical DOS drive without an Extended DOS Partition on the current drive

What this means Logical drives can only be created in an extended DOS partition, and none exists.

How to solve the problem Use FDISK to create an extended DOS partition.

Cannot delete Extended DOS Partition while logical drives exist

What this means Since logical drives require an extended DOS partition, you cannot delete the partition without first deleting all the logical drives.

How to solve the problem Use FDISK to delete the logical drives, remembering to backup the information they contain first.

Cannot DISKCOMP to or from an ASSIGNed or SUBSTed drive

What this means One of the drives referenced in the DISKCOPY command was created with the ASSIGN or SUBST command and DOS does not support this.

Cannot DISKCOMP to or from a network drive

What this means You are attempting to use DISKCOPY when one of the drives is a network drive and DOS does not support this action.

Cannot do binary reads from a device

What this means You cannot use the /b switch for binary mode when you are copying from a device.

How to solve the problem Eliminate the /b switch or use the /a switch to indicate ASCII.

Cannot edit .BAK file -- rename file

What this means DOS does not allow you to edit files with a .BAK extension since they are backup files created by EDLIN.

How to solve the problem Rename the file to create an extension other than .BAK.

Cannot find FORMAT.EXE

What this means The FORMAT command file is not accessible from the current directory and you are attempting to create a backup with an unformatted disk.

How to solve the problem Check to make sure the FORMAT file is on the current diskette. If it is not, insert the DOS diskette and re-execute. If you are using a hard

drive, make sure that PATH is set to the sub-directory where the DOS files are located.

Cannot find GRAPHICS profile

What this means DOS is looking in the current directory for GRAPHICS.PRO and cannot find it.

How to solve the problem Specify the path for this file.

Cannot find System Files

What this means System files were not in the root directory of the drive you tried to use.

How to solve the problem Use a different drive or insure that the system files are in the root directory of the drive you are using.

Cannot format an ASSIGNed or SUBSTed drive

What this means You attempted to use FORMAT with a drive created with ASSIGN or SUBST and DOS does not allow this.

Cannot FORMAT a Network drive

What this means DOS does not allow you to format a network drive.

How to solve the problem In order to format the drive, you must use the network software that came with the network. Contact your network administrator.

Cannot FORMAT nonremovable drive *x:*

What this means You are running backup with the /f switch for formatting, yet you are using the fixed disk as the target location for these files and DOS will not allow you to format the fixed disk.

How to solve the problem Consider using a removable disk for backup. If you want to use the fixed disk you must use the formatting already done to the disk and you cannot reformat it as part of backup.

Cannot JOIN a Network drive

What this means You cannot JOIN network drives.

Cannot LABEL a Network drive

What this means Drives which are shared on a network cannot be labeled with DOS.

How to solve the problem Use the network software to handle maintenance on network drives.

Cannot LABEL a SUBSTed or ASSIGNed drive

What this means If you have used a drive with SUBST or ASSIGN, you cannot write a label for the drive.

Cannot load COMMAND, system halted

What this means DOS is unable to reload COMMAND.COM.

How to solve the problem Reboot your system.

Cannot preform a cyclic copy

What this means You have specified the /s switch with XCOPY to copy all lower level subdirectories and you have specified one of these subdirectories as the target for the copy operation.

How to solve the problem Respecify XCOPY either without /s, or use a target location outside the current source copy range.

Cannot read file allocation table

What this means The disk is damaged and the file allocation table is not readable.

How to solve the problem Use your backup copy of the disk.

Cannot recover . entry, processing continued

What this means The current directory entry is defective.

How to solve the problem Try running CHKDSK with the /f option.

Cannot recover .. entry, Entry has a bad attribute (or link or size)

What this means Parent directory entry is defective.

How to solve the problem Run CHKDSK with the /f option.

Cannot RECOVER a Network drive

What this means The RECOVER command cannot be used on network drives.

Cannot setup expanded memory

What this means Your expanded memory card is not functioning.

How to solve the problem Check the seating of the card in its slot. Check with your computer dealer if this does not solve the problem.

Cannot specify default drive

What this means Different drives must be specified with the SYS command.

Cannot start COMMAND, exiting

What this means DOS is unable to start the command processor.

How to solve the problem Correct the PATH in AUTOEXEC.BAT or increase the number of files in FILES= within CONFIG.SYS.

Cannot SUBST a Network drive

What this means You cannot use SUBST with network drives.

Cannot SYS a Network Drive

What this means This command is invalid on a network.

Cannot use FASTOPEN for drive *x:*

What this means You are trying to use FASTOPEN on a drive other than a fixed-disk drive, on a network drive, or with more than four disks. None of these actions are allowed.

Cannot use PRINT - Use NET PRINT

What this means You cannot use PRINT on a network.

How to solve the problem Use the NET PRINT command or a print utility supplied with your network software.

Cannot XCOPY to a reserved device

What this means XCOPY will not work with the specified device.

How to solve the problem Check to be sure that you have not specified NULL or a character device. Use a file or a block device and reexecute the command.

CHDIR .. failed, trying alternate method

What this means CHKDSK was unable to return to a parent directory. DOS is starting over at the root directory.

CHKDSK not available on drive *X*

What this means You are trying to run CHKDSK on a file system that DOS cannot locate.

Code page not prepared

What this means Either the code page you selected is not prepared or it does not have the correct font to support your video mode.

How to solve the problem Use MODE CODEPAGE PREPARE. If DISPLAY.SYS is installed, check the DEVICE command in CONFIG.SYS to be sure it allows for additional fonts.

Code page *xxx* not prepared for all devices

What this means The code Page you selected is not supported on a device.

How to solve the problem Check device to see if it is online and if it supports switching code pages. Use MODE CODEPAGE PREPARE then CHCP.

Code page *xxx* not prepared for system

What this means CHCP is not able to select a code page.

How to solve the problem Install NLSFUNC if it is not installed. If you have used DEVICE in CONFIG.SYS, try MODE CODEPAGE PREPARE before retrying CHCP.

Code page operation not supported on this device

What this means Invalid code page and device combination specified.

Code page requested *xxx* is not valid for given keyboard code

What this means Code page and the keyboard code are not compatible.

Code page specified has not been designated

What this means KEYB has been used with an unrecognized code page.

How to solve the problem Use MODE CODEPAGE PREPARE before retrying KEYB.

Code page specified had not been prepared

What this means KEYB has been used with an unrecognized code page.

How to solve the problem USE MODE CODEPAGE PREPARE before retrying KEYB.

Code page specified is inconsistent with Invoked code page

What this means Code page specified with KEYB is inconsistent with selected code page for CON.

How to solve the problem Change the code page for CON with MODE CODEPAGE SELECT.

Code page specified is inconsistent with selected code page

What this means Code page specified with KEYB is inconsistent with selected code page for CON.

How to solve the problem Change the code page for CON with MODE CODEPAGE SELECT.

Code pages cannot be prepared

What this means Duplicate code page specified for device or tried to prepare more than the limit supported.

How to solve the problem Check DEVICE command in CONFIG.SYS to see how many code pages you can have and use MODE with /status to see the prepared code pages.

Compare error at OFFSET *X*

What this means The COMP command has detected differences in the files at the offset specified.

Compare error on disk side *s*, track *t*

What this means DISKCOMP detected a difference in the two disks.

Compare process ended

What this means A fatal error occurred during a compare operation.

Content of destination lost before copy

What this means Source file was overwritten before copy was completed.

How to solve the problem Use a backup copy of the file and reissue the copy operation.

Copy process ended

What this means DISKCOPY was not able to complete successfully.

How to solve the problem USE COPY or XCOPY to copy as many files as possible.

Current keyboard does not support this code page

What this means Keyboard code and code page are incompatible.

How to solve the problem Change the code page or KEYB.

Data error reading drive *x:*

What this means DOS is not able to read the data due to a defective disk.

How to solve the problem Try using Retry and if it fails try a complete new copy of the disk.

Device error during status

What this means DOS encountered an error condition with the device when performing a status check.

How to solve the problem Normally this problem involves code pages and might be due to any of the following:

A bad font file
Inability to support more code pages
No support for code pages
Improper preparation for code pages

Check to see if your device supports code page switching then check the DEVICE command in the CONFIG.SYS file to be sure that the syntax is correct.

Device error during prepare

What this means DOS was unable to prepare the device for code page switching.

How to solve the problem If the device does not support code page switching the problem cannot be resolved. Check the DEVICE statement in CONFIG.SYS for syntax errors. You can check to see if the device is prepared for code page switching or if it is already supporting as many code pages as possible. You might also try replacing the font file in the event that it is bad.

Device error during select

What this means DOS found an error with the device when it attempted to select it. Problem is likely to relate to code pages.

How to solve the problem See the previous two errors as the problem and correction techniques are the same.

Device error during write of font file to device

What this means DOS attempted to write a font file to the device and encountered an error. Code page problems are the likely cause.

How to solve the problem See device error during status.

Device or code page missing from font file

What this means Definition of code page for the device was not in the font file.

How to solve the problem Use MODE to specify a code page for the device.

Device *ddd* not prepared

What this means A code page was not prepared for the device.

How to solve the problem Use MODE to install a code page for the device.

Directory already exists

What this means You specified a directory with MD that already exists.

How to solve the problem Either use a new name for the directory you are attempting to establish or delete the old one if it is obsolete.

Directory is totally empty, no . or ..

What this means DOS is unable to find the required . and .. files in the directory.

How to solve the problem Delete the directory and recreate it.

Directory not empty

What this means You attempted to remove a directory while some files were still present.

How to solve the problem Either move to the directory and delete the files, or move the files somewhere else. You may then proceed to remove the directory.

Disk error reading (or writing) drive *x:*

What this means DOS could not read the disk data since the disk is probably defective.

How to solve the problem Retry several times, then attempt to make a copy of the disk. If unsuccessful, use your backup copy of the data.

Disk error reading (or writing) FAT *X*

What this means The file allocation table contains a bad sector.

How to solve the problem Copy the files or use CHKDSK with /f.

Disk full. Edits lost.

What this means You had attempted to save a file that required more space than what was available on the disk.

How to solve the problem Either abort the edit or delete some obsolete files to create room for the edit. If you chose the latter, you must perform the edit again.

Disk full error writing to BACKUP Log File

What this means No space available on log file disk.

How to solve the problem Press CTRL-BREAK, delete unnecessary data, or use a different disk.

Disk unsuitable for system disk

What this means The track used for system data is bad.

How to solve the problem Use a different disk.

Do not specify filename(s) Command format: DISKCOMP d: d: [/1][/8]

What this means Invalid switch used, or a filename specified along with a drive.

How to solve the problem Check the syntax of DISKCOMP, edit, and retry.

Do not specify filename(s) Command format: DISKCOPY d: d: [/1]

What this means Incorrect syntax of DISKCOPY.

How to solve the problem Correct syntax and retry.

Drive *x* already deleted.

What this means You tried to delete a drive that does not exist.

Drive types or diskette types not compatible

What this means You tried to use either DISKCOPY or DISKCOMP with drives of different type or size.

How to solve the problem Use XCOPY or COPY if you are copying, and COMP for the comparison operation when drives differ in type or size.

Duplicate file name or file not found

What this means File you are renaming could not be located, or you have chosen a new name for the file that is already used by another file.

How to solve the problem Check to be sure the file exists and the filename is unique.

Duplicate parameters not allowed

What this means You specified a parameter more than once.

Entry error

What this means The last command contains a syntax error.

How to solve the problem Correct the syntax and rerun the command.

Entry has a bad attribute (or link or size)

What this means DOS has encountered an invalid subdirectory entry.

How to solve the problem Run CHKDSK with the /f option.

Error during read of font file

What this means DOS found an error in the font file for the code page that you specified.

How to solve the problem Replace the font file.

Error in COUNTRY command

What this means COUNTRY command in CONFIG.sys contains a syntax error.

Errors found, F parameter not specified
Corrections will not be written to disk

What this means CHKDSK found errors on the disk but was unable to correct them since you did not use the /f switch.

How to solve the problem Rerun CHKDSK with /f.

Error in EXE file

What this means EXE file you attempted to use does not have a valid format for the version of DOS that you are using.

How to solve the problem Check to be sure that you are using the correct version of DOS and the .EXE file.

Error loading operating system

What this means DOS was unable to boot from your fixed disk.

How to solve the problem Boot the system from a diskette then use SYS to transfer a new copy of the system files to your fixed disk.

(Error occurred in environment variable)

What this means The value of the DIRCMD environment variable is not legal.

How to solve the problem Use the SET command to display and then change DIRCMD.

Error opening log file

What this means DOS was not able to open the log file used with the BACKUP command.

How to solve the problem Specify a log file location other than the target drive.

Error reading directory

What this means Bad sectors were found in the file allocation table.

How to solve the problem Discard the disk if it is a floppy disk or reformat the disk if it is a fixed disk, and restore the data from your most recent backup.

Error reading fixed disk

What this means FDISK tried five times but cannot read the fixed disk drive.

Error reading GRAPHICS profile

What this means DOS is unable to read GRAPHICS.PRO.

How to solve the problem Check the drive to insure that it is closed, then check the PATH statement.

Error reading MS-DOS system file

What this means DOS could not read the system files on the specified drive.

How to solve the problem Check the drive to see if it contains system files and transfer them if necessary.

Error reading/writing partition table

What this means The partition table could not be read.

How to solve the problem Use FDISK to reestablish the partitions and FORMAT them again.

Error Writing MS-DOS system file

What this means A bad disk caused an error in writing a system table.

How to solve the problem Use another disk.

Error writing directory (FAT or fixed disk)

What this means The file allocation table or directory contain bad sectors.

How to solve the problem Reformat the disk.

Error writing to device

What this means DOS was unable to handle the amount of data sent to the device.

How to solve the problem Send less data to the device.

EXEC failure

What this means DOS encountered an error when reading a command or FILES = contains lower number.

How to solve the problem Increase FILES= in CONFIG.SYS and reboot.

Extended DOS partition already exists

What this means You have attempted to create a DOS partition, but one already exists.

Failure to access code page font file

What this means DOS is unable to open the font file for the code page.

How to solve the problem Check the DEVICE command in CONFIG.SYS and check the font pathname.

Failure to access device: *xxx*

What this means Code page cannot be specified for the device since DOS cannot access the device.

How to solve the problem Check the spelling of the DEVICENAME.

Fastopen already installed

What this means You are attempting to install FASTOPEN and it is already installed.

FCB unavailable reading (or writing) drive *x*:

What this means The file control block is unavailable.

How to solve the problem Contact your computer dealer or a system programmer.

---

fc: cannot open *filename* - No such file or directory

What this means File or directory specified in the fc command does not exist.

How to solve the problem Check the spelling and location of the filename.

---

fc: *filename* longer than *filename*

What this means One file in a comparison operation is longer than the other.

---

fc: Incompatible switches

What this means fc switch options are not valid.

How to solve the problem Check the syntax of your entry. Also check to be sure that you did not use the /b and /a options together since binary and ASCII are not compatible.

---

fc: out of memory

What this means Insufficient memory is available to compare with fc.

How to solve the problem Remove TSR's from memory or exit from Shell.

File allocation table bad

What this means The file allocation table is bad. This is an indication that the disk is bad.

How to solve the problem Run CHKDSK with /f.

File allocation table bad drive *x*:

What this means Disk is not formatted properly for DOS.

How to solve the problem Run CHKDSK /f and reformat the disk if the error occurs again.

File *filename* canceled by operator

What this means The /t switch was used with the PRINT command.

File cannot be converted

What this means You are attempting to use the file with the exe2bin command and it is not in the correct format.

File cannot be copied into itself

What this means Both source and target filenames are the same.

How to solve the problem Change the target filename or its path.

File creation error

What this means There is insufficient space for the file or the file already exists and it is read-only.

How to solve the problem Delete files if the file does not exist. If the file exists, use ATTRIB to remove the read-only attribute.

File is READ-ONLY

What this means You are attempting to change a file that is read-only.

How to solve this problem If you want to modify the file use ATTRIB -r to remove the read-only attribute.

File MSDOS.SYS not found on specified drive

What this means The drive you specified does not contain system files.

How to solve the problem Specify another drive.

File name must be specified

What this means You attempted to use EDLIN without indicating a filename.

How to solve the problem Enter EDLIN with a filename.

File not found

What this means DOS could not find the filename.

How to solve the problem Check the spelling of the filename.

File not in PRINT queue

What this means You cannot remove the file from the print queue since it is not in the queue.

How to solve the problem Check the spelling of the filename.

FIND: Access Denied

What this means DOS will not allow you to access the file.

FIND: File not found

What this means DOS was unable to locate the file.

How to solve the problem Check the spelling of the filename and its extension.

FIND: Invalid number of parameters

What this means Too many or too few parameters were specified with FIND.

How to solve the problem Check the syntax of the statement.

FIND: Invalid Parameter

What this means One of the FIND switches is invalid.

How to solve the problem Check the syntax of the FIND command.

FIND: Read error in *filename*

What this means FIND is unable to read the specified filename.

How to solve the problem Check the spelling of the filename and its path or replace the file with a backup copy.

FIND: Syntax error

What this means There is an error in the entry of the FIND command.

How to solve the problem Correct the syntax problem.

First allocation unit is Invalid, entry truncated

What this means The directory contains an invalid pointer to data.

How to solve the problem Use CHKDSK with /f.

FIRST diskette bad or Incompatible

What this means The first diskette is not recognized and cannot be compared.

How to solve the problem Try using CHKDSK to see if the disk has been damaged.

Fixed backup device *x* is full

What this means The device specified can no longer be used for backups since it is full.

Fixups needed - base segment hex:

What this means A memory address must be specified in order to load the .EXE file.

How to solve the problem Check the documentation for the program you are attempting to load.

Font File contents Invalid

What this means DOS is unable to use the specified file as a font file.

How to solve the problem Check the spelling of the filename. Replace font file from a backup copy.

FOR cannot be nested

What this means The FOR command cannot be nested within another FOR command in a batch file.

Format not supported on drive *x*:

What this means FORMAT cannot be used with the specified drive.

How to solve the problem Check the DEVICE parameters.

Function not supported on this computer - x

What this means You do not have the proper device or adaptor attached to handle the request.

How to solve the problem Specify another device or adaptor.

General failure reading (or writing) drive *x*:

What this means The computer is unable to read (or write) to the disk. This is usually caused by a bad diskette, an unformatted disk, or a door not closed.

How to solve the problem Make sure your disk door is closed. If it is, format the diskette.

Has Invalid allocation unit, file truncated

What this means File directory contains an invalid pointer.

How to solve the problem Run CHKDSK with /f.

HMA not available: loading DOS low

What this means NO high memory is available for DOS and DOS ignores your request for loading into high memory.

How to solve the problem Check CONFIG.SYS for the line DEVICE=HIGHMEM.SYS. It must be present and after DOS=HIGH.

Illegal device name

What this means DOS does not recognize the device name specified.

How to solve the problem Check to be sure the device name is legal for the mode command and that it is spelled correctly.

Illegal device name - COM*x*

What this means An invalid COM port was specified.

How to solve the problem Check the syntax of your entry.

Incorrect APPEND version

What this means You attempted to use an APPEND command from an earlier version of DOS.

How to solve the problem Copy APPEND from your backup DOS disk.

Incorrect DOS Version

What this means You are attempting to run a DOS utility with an older version of DOS and they are not compatible.

How to solve the problem Use the correct version of DOS to run the command.

Incorrect DOS Version, use DOS 2.00 or later

What this means Certain DOS functions require a 2.00 or later release.

How to solve the problem Reboot your system with the correct release of DOS.

Incorrect number of parameters

What this means The entry in the command line does not have the correct number of parameters.

How to solve the problem Add or delete parameters in the command line.

Incorrect parameter

What this means You used the FASTOPEN command with an incorrect parameter.

How to solve the problem Check the syntax and reexecute.

Infinite retry on parallel printer timeout

What this means The printer is offline or not ready.

How to solve the problem Reset your printer. If this is not successful, reboot your system.

Infinite retry not supported on network printer

What this means Your system cannot support error detection on a network printer.

How to solve the problem Use a local printer or omit the retry specification.

Insufficient disk space

What this means You have attempted to write a file too large for the space available on the disk.

How to solve the problem Either use another disk of sufficient size for the file you are trying to save or delete some obsolete files on the disk, and save the file again.

Insufficient memory

What this means The command or program you are trying to execute needs more RAM than is available.

How to solve the problem If you have any TSRs loaded into memory, you will need to remove them.

Insufficient room in root directory. Erase files in root and repeat CHKDSK

What this means The CHKDSK command requires some memory in the root for recovered files.

How to solve the problem You will need to erase or move some files in the root directory to create space.

Insufficient space in version table for new entry

What this means Version table is full.

How to solve the problem Delete entries from this table.

Intermediate file error during pipe

What this means An error occurred in a temporary file during a pipe operation.

How to solve the problem Make sure there is space on the disk and that it is not write-protected.

Internal error

What this means An error occurred in a DOS utility.

How to solve the problem Try the utility again. Replace the utility if the problem persists.

Internal stack overflow System Halted

What this means A hardware interrupt occurred due to a lack of stacks.

How to solve the problem Alter CONFIG.SYS to allow more stacks, then restart DOS.

Invalid argument

What this means Command entry contains an invalid argument.

How to solve the problem Check Appendix F for the syntax of the command.

Invalid baud rate specified

What this means Baud rate specified is not valid.

How to solve the problem Use 110, 150, 300, 600, 1200, 2400, 4800, 9600, or 19200 as a baud rate.

Invalid characters in volume label

What this means The volume label you have specified contains invalid characters.

How to solve the problem Restrict your entry to only the alphabetic characters and the underscore symbol.

Invalid code page specified

What this means Code page number specified is not correct.

How to solve the problem Reenter with the correct code page number.

Invalid COMMAND.COM Insert COMMAND.COM disk in drive and strike any key when ready

What this means The COMMAND.COM is necessary, but is either bad or missing from the current disk.

How to solve the problem Either insert a disk with COMMAND.COM present or copy COMMAND.COM to the current diskette and execute the command again.

Invalid country code or code page

What this means The CONFIG.SYS file contains an invalid country code or code page number.

How to solve the problem Correct the CONFIG.SYS file and reboot DOS.

Invalid current directory

What this means The current disk contains an invalid directory.

How to solve the problem Try to copy the files to another disk.

Invalid date

What this means The date you have entered is in the wrong format.

How to solve the problem Use the format MM/DD/YY.

Invalid Date/Time

What this means The date or time you entered is in the wrong format.

How to solve the problem Enter the Time or Date again using the HH:MM:SS format for time or the MM-DD-YY format for year.

Invalid device

What this means You entered an invalid device name.

How to solve the problem Restrict your device name entries to NUL, AUX, CON, and PRN.

Invalid device parameters from device driver

What this means This message can display when a partition does not start on a track boundary and may have been caused by reformatting a disk previously formatted with an early release of DOS without first using FDISK. This message can also be caused by an error in a DEVICE or DRIVEPARM command in CONFIG.SYS.

How to solve the problem Check CONFIG.SYS for an invalid DEVICE or DRIVEPARM command. If that is not the problem, run FDISK and reformat before restoring data from backup.

Invalid directory

What this means Directory specified is not valid.

How to solve the problem Check the spelling or create the desired directory.

Invalid disk change reading (or writing) drive *x*:

What this means DOS needs the disk it was using before you changed the disk in the drive.

How to solve the problem Put the original disk back in the drive.

Invalid drive in search path

What this means Specified drive does not exist.

How to solve the problem Change the path specified or the PATH statement.

Invalid drive or filename

What this means The drive or filename you entered is not valid.

How to solve the problem Correct your entry.

Invalid drive specification

What this means The drive specification is invalid.

How to solve the problem Make sure that the drive you entered exists.

Invalid entry, please enter *x*

What this means You entered a value for FDISK that is not valid.

How to solve the problem Change your entry to conform to the limitations of FDISK.

Invalid entry, please press Enter

What this means The last character you are allowed to enter is %.

How to solve the problem Correct and reenter.

Invalid environment size specified

What this means The size specified with /e is incorrect.

How to solve the problem Enter a size between 160 and 32768.

Invalid file/directory entry

What this means The size of the filename cache is not correct.

How to solve the problem Use the default of 34 or specify a size between 10 and 999.

Invalid filename

What this means The filename that you entered is not valid.

How to solve the problem Check to be sure you have not used a reserved DOS name or made another mistake in following naming conventions.

Invalid keyboard code specified

What this means Entry used with the KEYB command is not valid.

How to solve the problem Reenter the command with the correct keyboard code.

Invalid keyword

What this means An invalid keyword has been used in the command.

How to solve the problem Use the /? option to display the correct command syntax.

Invalid number of parameters

What this means You either omitted an option or specified an extra one.

How to solve the problem Check the syntax for the command with /?.

Invalid parameter

What this means One of the parameters you have specified is invalid for the command.

How to solve the problem Make sure the parameter is legal for this command with the /? option.

Invalid parameter combination

What this means You may have put the parameters in the incorrect form for the command.

How to solve the problem Make sure you are using the correct form of the command.

Invalid parameter (or parameter combination)

What this means An invalid switch or parameter was used with the command.

How to solve the problem Check the command syntax with /?.

Invalid parameter(s)

What this means An invalid switch was used.

How to solve the problem Check the command syntax.

Invalid partition table

What this means An invalid entry exists in the partition table.

How to solve the problem Use FDISK to display the current partition information and make any needed corrections.

Invalid path

What this means The path specified is incorrect.

How to solve the problem Make sure that the pathname does not contain invalid characters and that it is within the length limit of 63 characters.

Invalid path, not directory, or directory not empty

What this means Directory cannot be removed because it is not empty, or it is not a valid directory.

How to solve the problem Correct the directory specification, or delete the files in the directory.

Invalid path (or file not found)

What this means Path or filename does not exist.

How to solve the problem Correct and reenter.

Invalid path or parameter

What this means The file or directory you specified does not exist.

How to solve the problem Use APPEND with a valid entry.

Invalid profile statement on line *x*

What this means The profile statement on the specified line contains a word which is not valid.

How to solve the problem Correct the specified line.

Invalid STACK parameter

What this means The STACK command in CONFIG.SYS contains an error.

How to solve the problem Edit CONFIG.SYS to make the correction and reboot.

Invalid sub-directory entry

What this means The subdirectory that you specified is not valid.

How to solve the problem Check the spelling and reenter.

Invalid switch

What this means The command line contains an invalid switch.

How to solve the problem Use /? to display the syntax, then make the correction.

Invalid switch type

What this means A duplicate switch, an invalid switch, or a switch that is not in the correct order was used.

How to solve the problem Use /? to check the syntax of the command.

Invalid syntax

What this means An error exists in the command syntax.

How to solve the problem Use /? to determine the correct syntax.

Invalid syntax on DISPLAY.SYS code page driver

What this means The DEVICE command for DISPLAY.SYS in CONFIG.SYS is invalid.

How to solve the problem Edit CONFIG.SYS and make the correction before rebooting.

Invalid syntax on PRINTER.SYS code page

What this means The DEVICE command for DISPLAY.SYS in CONFIG.SYS is invalid.

How to solve the problem Edit CONFIG.SYS and make the correction before rebooting.

Invalid time

What this means The time you have entered is in the wrong format.

How to solve the problem Use the format hh:mm:ss, as in 8:30:15 or 8:30.

Invalid version number format must be 3.20 - 9.99

What this means Version number is not valid.

How to solve the problem Use a number between 3.2 and 9.99.

Invalid Volume ID

What this means Volume label entered with FORMAT does not match the label on the fixed disk.

How to solve the problem Use VOL to determine the label on the fixed disk and reexecute the command if you want to format the fixed disk.

Is cross linked on allocation unit *x*

What this means Two files are allocated the same space on the disk according to the file allocation table.

How to solve the problem Execute CHKDSK with /f to truncate the file.

x is not a choice, Please enter y-z

What this means You cannot use the specified option with FDISK.

How to solve the problem Select from the options y-z, and run FDISK again.

KEYB has not been installed

What this means You are attempting to use an alternative keyboard code and KEYB is not installed.

How to solve the problem Install KEYB.

Label not found

What this means A GOTO command in a batch file is attempting to branch to a label that does not exist.

How to solve the problem Modify the batch file to add the label or correct the spelling in the label specified with the GOTO command to match an existing label.

Last backup disk not inserted Insert last backup disk in drive *x*: Strike any key when ready

What this means The final backup disk has not been entered.

How to solve the problem Insert the final disk and press any key.

*** Last file not backed up ***

What this means The backup process has run out of room for the final file or the file has an error on it.

How to solve the problem Back up the file separately.

Line too long

What this means When using EDLIN, a replace command would expand the line beyond its limit of 253 characters.

How to solve the problem Split the line into two lines and complete the edit.

List output is not assigned to a device

What this means PRINT output is assigned to a device that does not exist.

How to solve the problem Specify an existing device for PRINT.

Lock violation reading (or writing) drive *x:*

What this means Access was attempted for a file another program was using.

How to solve the problem Quit with an A for abort, or retry the access with an R.

---

x lost allocation units found in *y* chains Convert lost chains to files (Y/N)?

What this means CHKDSK found errors in the allocation of disk space to files.

How to solve the problem Type Y to have CHKDSK recover lost blocks of data and FDISK will create a file of this information that you can look at later.

LPT *x*: not rerouted

What this means The MODE command could not reroute the output for the parallel port.

How to solve the problem Check the syntax of your entry as well as the options you have selected.

Maximum number of logical DOS drives Installed

What this means You have attempted to create a logical drive when the maximum number possible have already been created.

Memory allocation error. Cannot load MS-DOS, system halted

What this means Your system was not able to boot DOS.

How to solve the problem Retry the boot operation. If this fails to correct the problem, boot your system from the backup copy of DOS.

Missing Parameter

What this means The command needs a parameter.

How to solve the problem. Check the form of the command.

MORE: Incorrect DOS version

What this means You have attempted to use MORE with a version of DOS before 2.0.

Must enter both /T and /N parameters

What this means You have used either the /T or /N parameter with FORMAT when the use of one requires that you use both.

How to solve the problem Either eliminate the switch you used or use both the /T and /N switch.

Must specify COM1, COM2, COM3, or COM4

What this means A serial port device must be specified.

How to solve the problem Specify COM1, COM2, COM3, or COM4.

Must specify destination line number

What this means You used the EDLIN copy (c) or move (m) command without a destination line number.

How to solve the problem Reenter the command with a line number.

Must specify ON or OFF

What this means The command requires an ON or OFF argument.

How to solve the problem Reenter the command with ON or OFF.

NLSFUNC already installed

What this means You have attempted to reload NLSFUNC and it is already loaded.

No Append

What this means No paths have been appended for searching for data files.

How to solve the problem Use the APPEND command if you wish to check for data in several locations.

No Appended directories

What this means A path was not used with the APPEND command.

How to solve the problem Reenter APPEND with directories if you want to check several locations for data files.

No code page has been selected

What this means A code page has not been selected.

How to solve the problem Use CHCP if you want to select a code page.

No drive specified

What this means DEVICE command was entered without a drive indicator.

How to solve the problem Modify the DEVICE command in CONFIG.SYS to indicate the physical drive.

No extended DOS Partition to delete

What this means You have attempted to delete a DOS partition using FDISK when no partition exists.

No files added (or replaced)

What this means The REPLACE command did not add or replace any files.

No files found *filename*

What this means REPLACE was unable to find the source or target file matching filename.

How to solve the problem Check the spelling of the filenames used with REPLACE.

No fixed disks present

What this means DOS cannot find the fixed disk you are trying to use or it is not operating properly.

How to solve the problem Check your drive specification.

No free file handles. Cannot start COMMAND.COM, exiting

What this means DOS has already opened the maximum number of files allowed and cannot open COMMAND.COM.

How to solve the problem Change CONFIG.SYS to increase the value in the FILES command and reboot.

No logical drives defined

What this means Logical drives have not been defined for your system.

How to solve the problem Use FDISK to create an extended DOS partition and logical drives if desired.

No paper error writing device *dev*

What this means The printer is off or out of paper.

How to solve the problem Check that the printer is on and that there is paper available for printing.

No partitions to delete

What this means You attempted to delete a partition with FDISK and none exist.

How to solve the problem Choose the correct FDISK option.

No partitions to make active

What this means You attempted to make a partition active and none exist.

How to solve the problem You must create a partition with FDISK before you can make it active.

No path

What this means No search path has been set.

How to solve the problem Use PATH to establish a search path.

No primary DOS partition to delete

What this means A primary DOS partition does not exist.

How to solve the problem Choose the correct FDISK option.

No room for system on Destination disk

What this means Lack of space prevents you from transferring the system files to the disk.

How to solve the problem Either change disks or erase/move files to create room.

No room in directory for file

What this means DOS was unable to add your file to the root directory of the disk.

How to solve the problem Delete something from the root directory to keep the number of files below the limit or store the file to a subdirectory where there are no file limitations.

No source drive specified

What this means You have attempted to execute a command which requires a source drive when none was specified.

How to solve the problem Check to insure you are using the correct form of the command.

No space left on device

What this means The target disk is full.

How to solve the problem Delete files or use another disk.

No space to create a DOS partition

What this means There is no room on the disk for another partition.

How to solve the problem Reduce the size of an existing partition, then use FDISK to create the new partition.

No space to create logical drive

What this means There is no space for creating a logical drive.

How to solve the problem Delete other logical drives.

No sub-directories exist

What this means You used the /s option with your command, but there are no subdirectories in the directory.

No such file in directory

What this means One or more files or directories does not exist.

How to solve the problem Check the spelling of all file and directory names.

No system on default drive

What this means You must use a disk with DOS on it for the default drive. The system must be present on the default drive.

How to solve the problem Add DOS to the default drive.

No target drive specified

What this means Command requires a target file location and none was specified.

Non-DOS disk error reading (or writing) drive *x:*

What this means DOS does not recognize the format of the current disk.

How to solve the problem Run CHKDSK and if that does not work, run format.

Non-System disk or disk error Replace and strike any key when ready

What this means Disk in the current drive is not a DOS disk.

How to solve the problem Replace the disk with a DOS disk.

*** Not able to back up (or restore) file ***

What this means There may be an error in the source file.

How to solve the problem Use CHKDSK to determine if an error exists.

Not enough memory

What this means Not enough memory is available to execute the command.

How to solve the problem Remove TSR's from memory or exit from current program before attempting to run the command again.

Not enough room to merge the entire file

What this means EDLIN transfer (t) command will not work because there is insufficient memory to hold the file.

How to solve the problem Write some files to disk or delete some files before retrying transfer (t).

Not found

What this means EDLIN search (s) or replace (r) was not able to find another occurrence of the string.

Not ready error reading (or writing) drive *x:*

What this means The specified device is not ready.

How to solve the problem Close the disk drive door or turn the printer on and execute the command again.

Only non-bootable partitions exist

What this means None of the partitions can boot DOS.

How to solve the problem Create a DOS partition with FDISK and copy DOS files to it.

Only partitions on drive 1 can be made active

What this means You are attempting to create a partition on a fixed disk other than drive 1 and this is not allowed.

Out of environment space

What this means No more space for environmental variables.

How to solve the problem Use /e with COMMAND to increase the space available for environment variables, or remove existing variables with SET.

Parameter value not in allowed range

What this means A parameter outside the acceptable range has been supplied.

How to solve the problem Check the syntax of the command and reenter.

Parameter value not allowed

What this means A parameter not within the allowable numeric range was specified.

How to solve the problem Check the syntax and reenter.

Parameters not compatible

What this means Multiple switches have been specified and they are not compatible.

How to solve the problem Check the syntax for the command.

Parameters not compatible with fixed disk

What this means Parameter specified is not compatible with the fixed disk.

How to solve the problem Check the command syntax.

Parameters not supported

What this means DOS does not support the parameter entered for this command.

Parameters not supported by Drive

What this means FORMAT command does not support the parameter on the drive specified.

How to solve the problem Check the syntax of FORMAT and reenter.

Partition selected (*x*) is not bootable, active partition not changed

What this means Partition you selected as the active partition cannot boot DOS.

How to solve the problem Choose a DOS partition for the active partition.

Parse Error *xx*

What this means Error in entry for a DOS utility, and DOS cannot locate COMMAND.COM for more detailed information.

How to solve the problem Check syntax and insure COMMAND.COM is available.

Path(*name*) too long

What this means Pathname exceeds DOS length limits.

How to solve the problem Try moving to the directory where the file is located before executing to stay with 64 character limit.

Path not found

What this means Pathname is invalid.

How to solve the problem Check spelling.

Primary DOS partition already exists

What this means You are trying to create a primary DOS partition when it already exists.

Printbox ID not in GRAPHICS profile

What this means You have not specified a print box size.

How to solve the problem Use /printbox:id to match the printbox statement in GRAPHICS.PRO.

Printer error

What this means DOS is unable to communicate with your printer.

How to solve the problem Check that printer is on and ready.

PRINT queue is empty

What this means No files are waiting to be printed.

PRINT queue is full

What this means No more files may be placed on the print queue without increasing its size. Use PRINT with /q to increase the queue size to a maximum of 32 files.

Printer type not in GRAPHICS profile

What this means DOS does not recognize your printer.

How to solve the problem Add description to the GRAPHICS command profile file.

Probable non-DOS disk Continue (Y/N)?

What this means Current disk is not recognized by the current version of DOS.

How to solve the problem Could be a non-DOS disk or a disk with an error in its file allocation table.

Processing cannot continue

What this means Cannot run CHKDSK with the current memory.

How to solve the problem Free memory before rerunning CHKDSK.

Profile statement out of sequence on line *x*

What this means Profile statement is in incorrect sequence.

How to solve the problem Resequence the profile statement.

Program too big to fit in memory

What this means Insufficient memory available to run the program.

How to solve the problem You may have to remove any currently loaded TSR's.

RAMDrive: Bad extended memory manager control chain

What this means DOS cannot install RAMDrive.

How to solve the probiem

Remove device drivers before RAMDrive in the CONFIG.SYS file to see if one is causing a problem.

RAMDrive: Error in extended memory allocation

What this means Memory error encountered.

How to solve the problem Memory diagnostic software must be run on your system to diagnose the problem.

RamDrive: Expanded Memory Manager not present

What this means The /a switch was used with RAMDrive.SYS, yet expanded memory manager was not installed.

How to solve the problem Load the expanded memory manager, then RAMDrive.SYS.

RAMDrive: Expanded Memory Status shows error

What this means RAMDrive.SYS could not be installed due to a memory error.

How to solve the problem Check the documentation for your expanded memory adapter card.

RAMDrive: Extended Memory Manager not present

What this means RAMDrive.SYS could not locate the expanded memory manager.

How to solve the problem Install HIMEM or XMS to manage expanded memory.

RAMDrive: Incorrect DOS version

What this means You need a DOS 3.00 or later to setup a RAM drive.

How to solve the problem Use DOS 3.0 or higher, then install RAMDrive.SYS.

RAMDrive: Insufficient memory

What this means There is not enough memory to create a RAM drive.

How to solve the problem Try removing active TSR's to create memory.

RAMDrive: Invalid parameter

What this means CONFIG.SYS entry for RAMDrive contains an error.

How to solve the problem Check the status of the DEVICE command for RAMDrive.SYS.

RAMDrive: I/O error accessing drive memory

What this means DOS detected an error in the memory it planned to use for the RAM drive, so the RAM drive was not created.

How to solve the problem Run the memory test for the memory DOS planned to use for the RAM drive.

RAMDrive: No extended memory available

What this means You cannot establish RAM drives on your computer since you do not have extended memory.

RATE and DELAY must be specified together RATE=*X* DELAY=*X*

What this means If you specify either RATE or DELAY, you must specify both.

Read error, COUNTRY.SYS

What this means DOS cannot read COUNTRY.SYS.

How to solve the problem COUNTRY.SYS file contains an error and must be replaced.

Read error in filename

What this means DOS was unable to read the file.

How to solve the problem Replace the file from a backup copy.

Read error, KEYBOARD.SYS

What this means DOS cannot read KEYBOARD.SYS.

How to solve the problem Replace file.

Read fault error reading drive *x:*

What this means DOS is unable to read device.

How to solve the problem Check to see that disk is in drive properly.

Requested logical drive size exceeds the maximum available space

What this means You have requested a logical drive larger than the space available.

How to solve the problem Limit the size of the new logical drive to the space available.

Requested Screen Shift out of range

What this means Display cannot be shifted any further.

Required font not loaded

What this means DISPLAY.SYS cannot find the needed font.

How to solve the problem Change DEVICE=DISPLAY.SYS to the maximum number of subfonts and restart DOS.

Required parameter missing

What this means Command requires a parameter that was not supplied.

How to solve the problem Check the command syntax.

Required profile statement missing before line *x*

What this means Profile statement omitted.

How to solve the problem Add the required profile statement.

Restore file sequence error

What this means Files have been restored from backups in the wrong order.

How to solve the problem Check the dates of backup disks and restore in the proper sequence.

Resynch failed. Files are too different

What this means Differences in files are too great for fc to proceed with compare.

Same drive specified more than once

What this means You attempted to use FASTOPEN more than once for the same drive.

Same parameter entered twice

What this means You entered /t, /n, /f, or /v twice, and it can only be used once.

How to solve the problem Reenter with no more than one occurrence of each switch.

SECOND disk bad or Incompatible

What this means The second disk is not the same format as the first.

How to solve the problem Try using CHKDSK to insure the disk is usable.

Sector not found error reading (or writing) drive *x*:

What this means Sector size larger than other devices loaded with CONFIG.SYS.

How to solve the problem Copy all files, then reformat.

Sector size too large in file *filename*

What this means Specified device driver cannot be used because it uses a larger sector size than other device drivers.

Seek error reading (or writing) drive *x*:

What this means DOS is unable to find the data on the disk.

How to solve the problem Check that the disk is in the drive securely.

Share already installed

What this means You are attempting to reinstall SHARE when it can only be installed once.

Sharing violation reading drive *x*:

What this means A program is trying to access a file used by another program.

How to solve the problem Type A for Abort or R for Retry. If unsuccessful you can wait and try again at a later time.

SMARTDrive: Bad extended memory manager control chain

What this means SMARTDRV.SYS cannot be installed because another program is not following extended memory management rules.

How to solve the problem Modify CONFIG.SYS to remove device drivers prior to SMARTDRV.SYS in an attempt to identify the problem.

SMARTDrive: Error in extended memory allocation

What this means A memory allocation error occurred and SMARTDRV.SYS cannot be installed.

How to solve the problem Consult the documentation for your extended memory adaptor card.

SMARTDrive: Expanded Memory Manager not present

What this means SMARTDRV could not be installed as there was no expanded memory manager installed.

How to solve the problem Install an expanded memory manager.

SMARTDrive: Extended Memory Manager not present

What this means SMARTDRV.SYS could not be installed without an extended memory manager installed.

How to solve the problem Install HIMEM or XMS before SMARTDRV.SYS.

SMARTDrive: Expanded Memory Manager Status shows error

What this means SMARTDRV was not installed due to a memory error.

How to solve the problem Run expanded memory diagnostic software.

SMARTDrive: I/O error accessing cache memory

What this means SMARTDRV.SYS was not installed.

How to solve the problem Add memory to your system.

SMARTDrive: Incorrect DOS version

What this means SMARTDRV.SYS only uses DOS 3.0 or higher.

SMARTDrive: Insufficient memory

What this means There is insufficient memory available to create SMARTDrive.

How to solve the problem You will need to either remove any active TSR's or acquire more memory.

SMARTDrive: Invalid parameter

What this means DEVICE command for SMARTDRV.SYS in the CONFIG.SYS file is incorrect.

How to solve the problem Check the syntax of the command.

SMARTDrive: No extended memory available

What this means There is insufficient memory to install SMARTDRV.SYS.

How to solve the problem Add memory or remove TSRs or other device drivers.

SMARTDrive: No hard drives on system

What this means SMARTDRV.SYS requires a hard disk and you do not have one.

SMARTDrive: Too many bytes per track on fixed drive

What this means Your hard disk and SMARTDRV.SYS are incompatible.

SORT: Incorrect DOS version

What this means You need to have a 2.00 version of DOS or later to execute this version of SORT.

SORT: Insufficient disk space

What this means The disk is full.

How to solve the problem Delete files or activate another drive.

SORT: Insufficient memory

What this means Not enough memory to run SORT.

How to solve the problem Try removing any active TSR's, device drivers, or other active programs.

Source and target drives are the same

What this means You cannot make the source and target drives the same.

SOURCE disk bad or Incompatible

What this means Bad disk or one that is incompatible with the source drive type.

How to solve the problem Put the disk in a compatible drive.

Source does not contain backup files

What this means You have used RESTORE with a disk that does not contain backup files.

How to solve the problem Check the disk in the drive and replace with the backup disk.

Source path required

What this means You used the REPLACE command without a source path.

How to solve the problem Reenter with the source path.

Specified COMMAND search directory bad

What this means The SHELL command in CONFIG.SYS is incorrect.

How to solve the problem Check location of COMMAND.COM and insure that DOS can locate it.

Specified drive does not exist, or is non-removable

What this means DISKCOPY and DISKCOMP cannot be used with fixed disk drives.

How to solve the problem Copy files with COPY or XCOPY and compare with COMP.

Specified entry was not found in the version table

What this means An unsuccessful attempt was made to delete an entry not in the version table.

How to solve the problem Display the entries with SETVAR.

Syntax error

What this means A command was entered incorrectly.

How to solve the problem Check the syntax and reenter.

Syntax errors in GRAPHICS profile

What this means Error in GRAPHICS PROFILE entry.

How to solve the problem Check syntax and reenter.

System files on the specified drive do not support a version table

What this means DOS files on the specified drive are not version 5.

How to solve the problem Version table only works with DOS 5.

Target cannot be used for backup

What this means Backup cannot be created on the current target disk.

How to solve the problem Reformat the target disk, or replace it.

Target disk bad or incompatible

What this means Target disk has bad tracks or is incompatible.

How to solve the problem Use CHKDSK to determine if the disk is bad. You may need to format the disk if it is new.

Target disk may be unusable

What this means The target disk is not usable.

How to solve the problem Format the disk or use CHKDSK to check it.

The last file was not restored

What this means There was insufficient room on the target disk or a bad target disk that prevented all files from being restored.

How to solve the problem Use CHKDSK to look at the disk.

The only bootable partition on drive 1 is already marked active

What this means The active partition must be bootable and on drive 1. There are no other partitions that meet these qualifications.

Too many drive entries

What this means You tried to specify a fifth drive with FASTOPEN, and only four are allowed.

Too many files open

What this means DOS could not open a .BAK file since there are already too many files open.

How to solve the problem Increase the value for FILES in CONFIG.SYS.

Too many name entries

What this means Total entries for a drive cannot exceed 999.

Too many open files

What this means File you requested could not be opened.

How to solve the problem Increase FILES= in CONFIG.SYS.

Too many parameters

What this means The command does not follow the established syntax rules.

How to solve the problem Check the command syntax.

Top level process aborted, cannot continue

What this means DOS cannot recover from the error.

How to solve the problem Reboot and use another disk.

Track 0 bad - disk unusable

What this means FORMAT cannot handle a bad track zero as it just encountered.

How to solve the problem Use another disk.

Tree past this point not processed

What this means TREE cannot continue due to an error.

Unable to access Drive *x*:

What this means FDISK cannot access the drive.

How to solve the problem Reboot your system and try again.

Unable to create directory

What this means DOS could not create the specified directory.

How to solve the problem Check to see if there is a file of the same name or a full disk.

Unable to create KEYB table in resident memory

What this means DOS could not create the country specific table specified.

How to solve the problem Check to see if there is adequate memory available.

Unable to perform refresh operation

What this means PRINTER.SYS cannot refresh the printer memory.

How to solve the problem Prepare and select the desired code page and set buffers = to 1 or greater.

Unable to reload with profile supplied

What this means GRAPHICS already loaded.

How to solve the problem Reboot and load again.

Unable to shift Screen

What this means Screen cannot be shifted any further.

Unable to write BOOT

What this means Boot record cannot be written on bad track 0.

How to solve the problem Use another disk.

Unexpected DOS Error *n*

What this means A DOS error occurred and the error number is provided.

Unrecognized command in CONFIG.SYS

What this means One of the commands in CONFIG.SYS is not valid.

How to solve the problem Check to insure the command is spelled correctly and is in the correct form.

Unrecoverable error in directory Convert directory to file (Y/N)?

What this means CHKDSK cannot correct a directory error.

How to solve the problem Type Y to make the bad directory a file.

Unrecoverable read error on drive *x* side *N*, track *N*

What this means DOS cannot read or write the data at this location.

How to solve the problem Check that the disk is in the drive properly.

Unrecoverable read (or write) error on drive *x*:

What this means DOS cannot read or write data.

How to solve the problem Check placement of the disk in the drive.

Version table is corrupt

What this means Version table is not usable.

How to solve the problem Replace the system tables.

Volume label does not match

What this means You cannot enter the incorrect volume label.

How to solve the problem Correct your entry.

Warning- directory full

What this means Root directory is too full to use RECOVER.

How to solve the problem Delete some unneeded files.

Write failure, disk unusable

What this means SYS command was not executed successfully due to a critical disk error.

How to solve the problem Reexecute SYS with another disk.

Write fault error writing drive *x*:

What this means DOS cannot write to the specified device.

How to solve the problem Check that the device is ready, that is, a disk drive has a properly inserted disk with the drive door closed and a printer in online and ready.

Write protect error Format terminated

What this means You have attempted to Format a disk which is write protected.

How to solve the problem Either change to a different disk or remove the write protection if the disk is obsolete.

Write protect error writing drive *x*:

What this means You have attempted to write to a disk which is write protected.

How to solve the problem Either change to a different disk or remove the write protection if you wish to write to that disk.

Key Summary

Table C-1: General Use Keys

Key	Description
F1	Displays Help information
F3	Exits the DOSSHELL and displays the command prompt
F10	Selects the menu at top of screen
ALT	Selects the menu at top of screen
ALT + F4	Exits the DOSSHELL and displays the command prompt
SHIFT + F9	Saves the DOSSHELL and displays the command prompt
End	Moves to the end of a line
CTRL + End	Moves to the end of a list
ENTER	Finalizes commands
ESC	Cancels the current request
Home	Moves to the beginning of a line
CTRL + Home	Moves to the beginning of a list
Letter	Moves to the next item in a list that starts with the requested letter
Page Up	Moves to the previous window
Page Down	Moves to the next window
Tab	Moves to the next section
SHIFT + Tab	Moves to the previous section
Up and Down Arrows	Scroll up and down in a list

Table C-2: Active Task List Keys

Key	Description
ALT + ESC	Moves to the next application
SHIFT + ALT + ESC	Moves to the previous application
ALT + TAB	Toggles between two applications
CTRL + ESC	Starts the DOSSHELL from an application
SHIFT (Held Down)	Cycles backward through the application
SHIFT + ENTER	Starts a program and adds it to the Task List
ALT + Letter CTRL + Letter SHIFT + Letter	Completes a personalized application shortcut

Table C-3: Program List Keys

Key	Description
F2	Copies programs in the Program List when you invoke the File Copy command
SHIFT + CTRL + ENTER	Starts a program in the Program List and opens a selected file in the File List
DEL	Deletes the selected program or group

Table C-4: File List and Selection Keys

Key	Description
F7	Moves the selected file from one directory to another
F8	Copies the selected file from one directory to another
F9 (when viewing file)	Displays the content of a file (Switches between ASCII + Hexadecimal display)
CTRL + SHIFT + ENTER	Starts a program in the Program List and opens a selected file in the File List
CTRL + /	Selects all files in a list

Key	Description
CTRL + \	Cancels file selection
SHIFT + F8	Toggle switch for the add mode
SHIFT + Up Arrow	Adds the previous file in the File List to the selection
SHIFT + Down Arrow	Adds the next file in the File List to the selection
SHIFT + PG UP	Adds files in the previous window's File List to the selection
SHIFT + PG DN	Adds files in the next window's File List to the selection
SHIFT + Spacebar (in Add Mode)	Selects the group of files between the previous file and the cursor
Del	Deletes the selected file
Spacebar	Adds the file at the cursor location to the selection

Table C-5: Directory Tree Keys

Key	Description
Up and Down Arrows	Move the cursor up and down to next directory level
* (asterisk)	Displays all directories
CTRL + * (CTRL + asterisk)	Displays all directories in the tree
+ (plus)	Displays one level of directories below the selected directory
- (minus)	Hides the directories below the selected directory

Table C-6: Drive Selection Keys

Key	Description
F5	Redisplays the Directory Tree and File List
CTRL + Drive	Reads the drive and displays directories
SHIFT + F5	Redraws the screen without updating the file list
ENTER	Displays the contents of the directory
Spacebar	Displays the directories on the selected drive

Table C-7: EDLIN Keys

Entry	Action	Sample Entry
Line #	Edits Line	10 Edits Line 10
A	Appends Line	5A Appends 5 Lines
B	Used when starting EDLIN to edit files containing CTRL+Z	EDLIN file 1B
C	Copies Lines	3,5C Copies Lines 3-5
D	Deletes Lines	1,5D Deletes Lines 1-5
E	Ends and Saves File	E Saves File & Ends
I	Inserts Line	I Allows you to begin inserting at the current line
L	Lists Line	1,10L List Lines 1-10
M	Moves Lines to another Location	2,6,7M Moves lines 2-6 to line 7
P	Page	1,5P Displays lines 1-5
Q	Quits not Saving Changes	Q Quits and ignores changes
R	Replaces Lines	ROldtext,Newtext
S	Searches for Text	SText
T	Transfers Data to a File	Ta:savethis
W	Writes out Lines	10W Writes 10 lines

Table C-8: Edit Keys

Keys	Description
F1	Provides help on menus and commands
SHIFT + F1	Displays information on getting started with the editor
CTRL + Q,F	Searches for text
F3	Repeats the search
CTRL + Ins	Copies to clipboard
CTRL + Q,Y	Deletes to the end of the line

Keys	Description
CTRL + Y	Deletes current line
Del	Erases text
SHIFT + Del	Deletes text
SHIFT + Left or Right Arrow Key	Selects character
SHIFT + Up or Down Arrow Key	Selects line
SHIFT + CTRL + Arrow	Selects word
Home , CTRL + N	Inserts line above cursor position
End + ENTER	Inserts line below cursor position
Ins	Toggles between insert and overwrite
SHIFT + Ins	Copies from the clipboard

Table C-9: Help Keys

Key	Description
F1	Invokes Help
SHIFT + F1	Displays information on getting started with the editor
ALT + H	Displays Help menu
Tab	Moves to the next Help screen
SHIFT + Tab	Moves to the previous Help screen
ALT + F1	Views the previous Help topic
CTRL + F1	Views the next topic in the Help file
SHIFT + CTRL + F1	Views the previous topic in the Help file
Letter	Moves to the next Help topic that begins with that letter
SHIFT + Letter	Moves to the previous Help topic that begins with that letter
ESC	Leaves Help

Table C-10: Cursor Movement and Scrolling Keys

Key	Description
Up Arrow	Moves cursor up
Down Arrow	Moves cursor down
CTRL + Up Arrow	Scrolls up
CTRL + Down Arrow	Scrolls down
Left Arrow	Moves one character to the left
Right Arrow	Moves one character to the right
CTRL + Left Arrow	Moves one word to the left
CTRL + Right Arrow	Moves one word to the right
Home	Moves to the indent on the present line
CTRL + Q,S	Moves to the beginning of the present line
CTRL + ENTER	Moves to the beginning of the next line
F6	Moves to the next window
ALT + Plus (+)	Increases Window size
ALT + Minus (-)	Decreases window size
CTRL + Q,E	Moves to the top of a window
CTRL + Q,X	Moves to the bottom of a window
PG UP	Moves one page up
PG DN	Moves one page down
CTRL + PG UP	Scrolls left
CTRL + PG DN	Scrolls right
CTRL + Q,0-3	Goes to a bookmark
CTRL + K,0-3	Sets bookmarks

Table C-11: Editing Keys

Key	Description
SHIFT + Ins	Pastes what is in the clipboard
End + ENTER	Inserts a line below cursor location
Home,CTRL + N	Inserts a line above cursor location
CTRL + P,CTRL +Character	Inserts special character
Backspace	Deletes the character behind the cursor
Del	Deletes character that the cursor is on OR deletes selected text
CTRL + T	Deletes the word the cursor is on
SHIFT + Tab	Deletes spaces at the beginning of selected lines
Ins	Toggles between insert and overwrite
CTRL + Ins	Copies text to the clipboard
SHIFT + Del	Copies text to the clipboard and then deletes it
CTRL + Y	Copies line to the clipboard and then deletes it
CTRL + Q,Y	Copies to end of line to the clipboard and then deletes it
CTRL + L	Repeats search for same text
CTRL + Q,F	Searches for text
CTRL + Q,A	Replaces text
SHIFT + Left Arrow	Selects the character to the left
SHIFT + Right Arrow	Selects the character to the right
SHIFT + Up Arrow	Selects the line above the current line
SHIFT + Down Arrow	Selects the present line
SHIFT + CTRL + Left Arrow	Selects the word to the left
SHIFT + CTRL + Right Arrow	Selects the word to the right
SHIFT + CTRL + Home	Selects to the beginning of the file
SHIFT + CTRL + End	Selects to the end of the file
SHIFT + PG UP	Selects one page up
SHIFT + PG DN	Selects one page down

ASCII Codes

Figure A-1: ASCII Codes and Descriptions

ASCII Code	Character	Description of Character
0	Null	Null
1	☺	Smiling face
2	☻	Smiling face, reversed
3	♥	Heart suit symbol
4	♦	Diamond suit symbol
5	♣	Club suit symbol
6	♠	Spade suit symbol
7	•	Bullet
8	◘	Bullet, reversed
9	○	Open circle
10	◙	Open circle, reversed
11	♂	Male symbol
12	♀	Female symbol
13	♪	Musical note
14	♫	Double musical note
15	☼	Sun symbol
16	►	Forward arrow indicator
17	◄	Back arrow indicator
18	↕	Up–down arrow
19	‼	Double exclamation points
20	¶	Paragraph symbol
21	§	Section symbol
22	▬	Solid horizontal rectangle
23	↨	Up–down arrow, perpendicular
24	↑	Up arrow
25	↓	Down Arrow
26	→	Right arrow
27	←	Left arrow
28	∟	Right angle symbol
29	↔	Left–right symbol
30	▲	Solid triangle
31	▼	Solid triangle inverted
32		Space
33	!	Exclamation point
34	"	Double quotes
35	#	Pound sign
36	$	Dollar sign
37	%	Percent
38	&	Ampersand
39	’	Close quote
40	(	Open parenthesis
41	)	Close parenthesis
42	*	Asterisk
43	+	Plus sign
44	,	Comma
45	–	Minus sign
46	.	Period
47	/	Slash
48	0	Zero
49	1	One
50	2	Two
51	3	Three
52	4	Four
53	5	Five
54	6	Six
55	7	Seven
56	8	Eight
57	9	Nine
58	:	Colon
59	;	Semicolon
60	<	Less than
61	=	Equal sign
62	>	Greater than
63	?	Question mark
64	@	At sign
65	A	A, uppercase
66	B	B, uppercase
67	C	C, uppercase
68	D	D, uppercase
69	E	E, uppercase
70	F	F, uppercase
71	G	G, uppercase
72	H	H, uppercase
73	I	I, uppercase
74	J	J, uppercase
75	K	K, uppercase
76	L	L, uppercase
77	M	M, uppercase
78	N	N, uppercase
79	O	O, uppercase
80	P	P, uppercase
81	Q	Q, uppercase
82	R	R, uppercase
83	S	S, uppercase
84	T	T, uppercase
85	U	U, uppercase
86	V	V, uppercase
87	W	W, uppercase
88	X	X, uppercase
89	Y	Y, uppercase
90	Z	Z, uppercase
91	[	Open bracket
92	\	Backslash
93	]	Close bracket
94	^	Caret
95	_	Underscore
96	‘	Open single quote
97	a	a, lowercase

Figure A-1: ASCII Codes and Descriptions (Continued)

ASCII Code	Character	Description of Character
98	b	b, lowercase
99	c	c, lowercase
100	d	d, lowercase
101	e	e, lowercase
102	f	f, lowercase
103	g	g, lowercase
104	h	h, lowercase
105	i	i, lowercase
106	j	j, lowercase
107	k	k, lowercase
108	l	l, lowercase
109	m	m, lowercase
110	n	n, lowercase
111	o	o, lowercase
112	p	p, lowercase
113	q	q, lowercase
114	r	r, lowercase
115	s	s, lowercase
116	t	t, lowercase
117	u	u, lowercase
118	v	v, lowercase
119	w	w, lowercase
120	x	x, lowercase
121	y	y, lowercase
122	z	z, lowercase
123	{	Open brace
124	|	Bar
125	}	Close brace
126	~	Tilde
127	DEL	Delete
128	Ç	C cedilla, uppercase
129	ü	u umlaut, lowercase
130	é	e acute, lowercase
131	â	a circumflex, lowercase
132	ä	a umlaut, lowercase
133	à	a grave, lowercase
134	å	a ring, lowercase
135	ç	c cedilla, lowercase
136	ê	e circumflex, lowercase
137	ë	e umlaut, lowercase
138	è	e grave, lowercase
139	ï	i umlaut, lowercase
140	î	i circumflex, lowercase
141	ì	i grave, lowercase
142	Ä	A umlaut, uppercase
143	Å	A ring, uppercase
144	É	E acute, uppercase
145	æ	ae diphthong, lowercase
146	Æ	AE diphthong, uppercase
147	ô	o circumflex, lowercase
148	ö	o umlaut, lowercase
149	ò	o grave, lowercase
150	û	u circumflex, lowercase
151	ù	u grave, lowercase
152	ÿ	y umlaut, lowercase
153	Ö	O umlaut, uppercase
154	Ü	U umlaut, uppercase
155	ø	o slash, lowercase
156	£	British pound sterling symbol
157	¥	Yen sign
158	Pt	Peseta sign
159	*f*	Guilder
160	á	a acute, lowercase
161	í	i acute, lowercase
162	ó	o acute, lowercase
163	ú	u acute, lowercase
164	ñ	n tilde, lowercase
165	Ñ	N tilde, uppercase
166	ª	Feminine ordinal indicator
167	º	Masculine ordinal indicator
168	¿	Question mark inverted
169	⌐	Start of line symbol
170	¬	End of line symbol/Logical NOT
171	½	One half
172	¼	One quarter
173	¡	Exclamation point, inverted
174	«	Left angle quotes
175	»	Right angle quotes
176	▒	Solid fill character, light
177	▓	Solid fill character, medium
178	▓	Solid fill character, heavy
179	│	Center vertical box bar
180	┤	Right box side
181	╡	Right box side, double joins single
182	╢	Right box side, single joins double
183	╖	Upper single right double box corner
184	╕	Upper double right single box corner
185	╣	Right box side, double
186	║	Center vertical box bar double
187	╗	Upper right box corner double
188	╝	Lower right box corner double
189	╜	Lower single right double box corner
190	╛	Lower double right single box corner
191	┐	Upper right box corner
192	└	Lower right box corner
193	┴	Lower box side
194	┬	Upper box side
195	├	Left box side

Figure A-1: ASCII Codes and Descriptions (Continued)

ASCII Code	Character	Description of Character
196	─	Center horizontal box bar
197	┼	Center box intersection
198	╞	Left box side, double joins single
199	╟	Left box side, single joins double
200	╚	Lower left box corner, double
201	╔	Upper left box corner, double
202	╩	Lower box side, double
203	╦	Upper box side, double
204	╠	Left box side, double
205	═	Center horizontal box bar double
206	╬	Center box intersection, double
207	╧	Lower box side, single joins double
208	╨	Lower box side, double joins single
209	╤	Upper box side, single joins double
210	╥	Upper box side, double joins single
211	╙	Lower single left double box corner
212	╘	Lower double left single box corner
213	╒	Upper double left single box corner
214	╓	Upper single left double box corner
215	╫	Center box intersection, vertical double
216	╪	Center box intersection, horizontal double
217	┘	Lower right box corner
218	┌	Upper left box corner
219	▉	Solid fill character
220	▄	Solid fill character, lower half
221	▌	Solid fill character, left half
222	▐	Solid fill character, right half
223	▀	Solid fill character, upper half
224	α	Alpha, lowercase
225	ß	Beta, lowercase
226	Γ	Gamma. uppercase
227	π	Pi, lowercase
228	Σ	Sigma, uppercase
229	σ	Sigma, lowercase
230	µ	Greek mu, lowercase
231	τ	Tau, lowercase
232	Φ	Phi, uppercase
233	Θ	Theta, uppercase
234	Ω	Omega, uppercase
235	δ	Delta, lowercase
236	∞	Infinity
237	ϕ	Phi, lowercase
238	∈	Epsilon, lowercase
239	∩	Join symbol
240	≡	Is the same as symbol
241	±	Plus or minus sign
242	≥	Greater than symbol
243	≤	Less than symbol
244	⌠	

ASCII Code	Character	Description of Character
245	⌡	
246	÷	Division sign
247	≈	Approximate symbol
248	°	Degree symbol
249	∙	Center dot
250	·	Small center dot
251	√	Square root symbol
252	ⁿ	n superscript
253	²	Two superscript
254	■	Square bullet
255		Null

Sample Code Pages

Figure E-1: Code Page 437 (United States)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p	Ç	É	á	▒	└	╨	α	≡
-1	☺	◄	!	1	A	Q	a	q	ü	æ	í	▒	┴	╤	β	±
-2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▒	┬	╥	Γ	≥
-3	♥	!!	#	3	C	S	c	s	â	ô	ú	▒	├	╙	π	≤
-4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
-5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
-6	♠	▬	&	6	F	V	f	v	å	û	ª	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	∙
-A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{	ï	¢	½	╗	╦	█	δ	√
-C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	▄	∞	n
-D	♪	↔	-	=	M	]	m	}	ì	¥	¡	╜	═	▌	ϕ	2
-E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	»	┐	╧	▀	∩	

Figure E-2: Code Page 850 (Multilingual)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	ð	Ó	-
-1	☺	◄	!	1	A	Q	a	q		ü	æ	í	▒	┴	Ð	β	±
-2	☻	↕	"	2	B	R	b	r		é	Æ	ó	▒	┬	Ê	Ô	=
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	Ë	Ò	¾
-4	♦	¶	$	4	D	T	d	t		ä	ö	ñ	┤	─	È	õ	¶
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	Á	┼	ı	Õ	§
-6	♠	▬	&	6	F	V	f	v		å	û	ª	Â	ã	Í	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ù	º	À	Ã	Î	þ	¸
-8	◘	↑	(	8	H	X	h	x		ê	ÿ	¿	©	╚	Ï	Þ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ö	®	╣	╔	┘	Ú	¨
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Û	·
-B	♂	←	+	;	K	[	k	{		ï	ø	½	╗	╦	■	Ù	¹
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▬	ý	³
-D	♪	↔	-	=	M	]	m	}		ì	Ø	¡	¢	═	¦	Ý	²
-E	♫	▲	.	>	N	^	n	~		Ä	×	«	¥	╬	Ì	´	■
-F	☼	▼	/	?	O	_	o	⌂		Å	ƒ	»	┐	¤	▬	′	

Figure E-3: Code Page 860 (Portugal)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	╨	α	≡
-1	☺	◄	!	1	A	Q	a	q		ü	À	í	▒	┴	╤	β	±
-2	☻	↕	"	2	B	R	b	r		é	È	ó	▒	┬	╥	Γ	≥
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	╙	π	≤
-4	♦	¶	$	4	D	T	d	t		ã	õ	ñ	┤	─	╘	Σ	⌠
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	╡	┼	╒	σ	⌡
-6	♠	▬	&	6	F	V	f	v		Á	Ú	ª	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ù	º	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x		ê	Ì	¿	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y		Ê	Õ	Ò	╣	╔	┘	Θ	•
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{		Ì	¢	½	╗	╦	■	δ	√
-C	♀	∟	,	<	L	\	l	|		Ô	£	¼	╝	╠	▄	∞	n
-D	♪	↔	-	=	M	]	m	}		ì	Ù	¡	╜	═	▌	ϕ	2
-E	♫	▲	.	>	N	^	n	~		Ã	Pt	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂		Â	Ó	»	┐	╧	▀	∩	

Figure E-4: Code Page 863 (Canada-French)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	¦	▒	└	╨	α	≡
-1	☺	◄	!	1	A	Q	a	q		ü	È	´	▒	┴	╤	β	±
-2	☻	↕	"	2	B	R	b	r		é	Ê	ó	▒	┬	╥	Γ	≥
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	╙	π	≤
-4	♦	¶	$	4	D	T	d	t		Â	Ë	¨	┤	─	╘	Σ	⌠
-5	♣	§	%	5	E	U	e	u		à	Ï	˘	╡	┼	╒	σ	⌡
-6	♠	▬	&	6	F	V	f	v		¶	û	³	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ú	˙	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x		ê	¤	Î	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ô	⌐	╣	╔	┘	Θ	•
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{		ï	¢	½	╗	╦	■	δ	√
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▄	∞	n
-D	♪	↔	-	=	M	]	m	}		=	Ù	¾	╜	═	▌	ϕ	2
-E	♫	▲	.	>	N	^	n	~		À	Û	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂		§	ƒ	»	┐	╧	▀	∩	

Figure E-5: Code Page 865 (Norway)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	▒	└	╨	α	≡
-1	☺	◄	!	1	A	Q	a	q		ü	æ	í	▒	┴	╤	β	±
-2	☻	↕	"	2	B	R	b	r		é	Æ	ó	▒	┬	╥	Γ	≥
-3	♥	!!	#	3	C	S	c	s		â	ô	ú	│	├	╙	π	≤
-4	♦	¶	$	4	D	T	d	t		ä	ö	ñ	┤	─	╘	Σ	⌠
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	╡	┼	╒	σ	⌡
-6	♠	‒	&	6	F	V	f	v		å	û	ª	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ú	º	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x		ê	ÿ	¿	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ö	⌐	╣	╔	┘	Θ	•
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{		ï	ø	½	╗	╦	█	δ	√
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▄	∞	n
-D	♪	↔	-	=	M	]	m	}		ì	Ø	¡	╜	═	▌	ϕ	2
-E	♫	▲	.	>	N	^	n	~		Ä	Pt	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂		Å	ƒ	¤	┐	╧	▀	∩	

DOS Commands

This appendix provides a comprehensive reference to all the DOS commands. In this appendix you will find:

- Commands that you can enter at the DOS prompt
- Commands that you can place in a batch file
- Commands that you can use in a CONFIG.SYS file

The information for each command provides a complete picture of the command and how it is used. For each command, you will find sections describing what the command does, how you enter the command, customization options, an example of how you might use the command and any miscellaneous notes that are appropriate for the command.

DOS commands are customized using either switches or parameters. Switches start with a / character and usually provide optional features for the command. Parameters may define the data that you wish to use with the command or other customizing options. When a parameter or switch uses a vertical bar as in /x:on | off, it means you can use one or the other, but not both. Optional switches and parameters are enclosed in square brackets. Italicized text represents characters or numbers that you must provide while the non-italicized text is entered as displayed. In addition to the parameters and switches listed for individual commands, all DOS commands and batch file commands take the /? switch, which displays an abbreviated syntax for the command. This provides the same output as the HELP command.

APPEND

What this command does Establishes a search path for data files that extends beyond the current directory. When used without parameters or switches, APPEND displays the current search path for these files. Any file that does not contain a program instruction is considered a data file.

How to enter this command APPEND can take several forms since the command supports the use of both parameters and switches.

```
APPEND [/x] [/e]
APPEND ;
APPEND
APPEND [pathname1][;pathname2][;pathname3]....[;pathnameN] [x:onloff][/path:onloff]
```

How to customize this command

Parameters

No parameter entry	Displays the current appended directory for data files.
;	Cancels any appended directory list previously added.
[*pathname1*][;*pathname2*]	Selects pathnames to append to the current directory as a search path for data files.

Switches

/e	Assigns the appended directories to the environmental variable APPEND. You can display the contents of this variable with SET. If you use this, it must be in the first APPEND command used in a DOS session.
/path:onloff	Selects whether DOS searches appended directories for data files when you include the pathname for the data file (/path:on) or whether DOS only searches appended directories when a filename and no directory information is provided. /path:on is the default.
/x:onloff	Selects whether DOS searches appended directories for executable files. /x:off, which only looks for executable files in the current directory, is the default setting. With /x:on, or with /x in the first APPEND command of the current DOS session, DOS searches current and appended directories for executable files.

How to use this command If you were working with a program such as 1-2-3, for example, and wanted to be able to access files in C:\123\BUDGET and D:\FINANCE, in addition to the current directory, you would enter the following:

```
APPEND C:\123\BUDGET;D:\FINANCE
```

Notes APPEND with the /x:on switch is similar to the PATH statement used to specify other directories to check for program files. The directories selected with APPEND are searched before the directories selected with the PATH command.

APPEND may not be an appropriate choice when you are working with a word processor. Since word processors frequently create new files rather than modify the originals, any file from an appended directory will have its new version written to the current directory even though the original copy of the file was in the appended directory.

Command type MS-DOS external command with network support.

ASSIGN

What this command does Assigns a different drive letter to an existing drive. This is especially useful when you are using a program that looks for files on drive A when the files are stored in drive C.

How to enter this command

ASSIGN [*x*=*y*][/status]

How to customize this command

Parameters

No parameter entry	Resets assigned drives to their original drive.
x	Selects the drive to reassign to another drive. The drive letter does not have a colon after it.
y	Selects the drive that the assigned drive letter will refer to. The drive letter does not have a colon after it.

Switch

/status or /s	Displays the current assigned drives.

How to use this command If, for example, you were using a program that expected files to be on drive A when you had stored them on drive C, you would enter this command:

```
ASSIGN A=C
```

Notes Do not use ASSIGN to assign the letter of the drive where DOS has COMMAND.COM stored.

Do not use the ASSIGN command with the BACKUP, DISKCOPY, FORMAT, JOIN, LABEL, PRINT, RESTORE, and SUBST commands. SUBST is usually a better command to use, since you can assign the drive to another drive and directory. You cannot assign a drive while another program is using it.

Command type MS-DOS external command with network support.

ATTRIB

What this command does Displays or changes the attributes assigned to files.

How to enter this command

ATTRIB [+rl-r] [+al-a] [+sl-s] [+hl-h] [*pathname*] [/s]

How to customize this command

Parameters

No parameter entry	Displays the file attributes of all files in the current directory.
pathname	Selects the files to list or change the attributes. This command only uses the current directory or the directory specified by the pathname unless /s is used.

Switches

+rl-r	Adds or removes a read-only attribute from the selected files. Files with this attribute cannot be modified.
+al-a	Adds or removes an archive attribute from the selected files. Commands such as BACKUP, RESTORE, and XCOPY can use this attribute to determine which files the command uses. An archive attribute is also added when you modify the file.
+sl-s	Adds or removes a system attribute from the selected files.
+hl-h	Adds or removes a hidden attribute from the selected files. Hidden files do not appear in file lists, such as those produced by the DIR command.
/s	Tells DOS to use all directories on the current drive for this command.

How to use this command You can use this command to change attributes of files. For example, you could make files read-only so you do not accidentally delete them. To do this for your .EXE files, you would enter:

```
ATTRIB +R *.EXE
```

Since you did not include the /s switch, only the .EXE files in the current directory become read-only. You can also use this command to find a file when you do not know where it is located, as in:

```
ATTRIB CASTLE.BAS /S
```

This checks all subdirectories on the current drive for a file called CASTLE.BAS. When it finds the file, DOS displays its full path, as well as its currently set attributes.

Command type MS-DOS external command with network support.

BACKUP

What this command does Copies one or more files to another disk or diskettes in special backup files. Unlike DISKCOPY, the source disk and destination disk do not have to be the same size.

How to enter this command

BACKUP [*pathname1*] [*pathname2*][/s][/m][/a][/f:*size*][/d:*date*][/t:*time*][/l[:*pathname*]]

How to customize this command

Parameters

pathname1	Selects the source of the files you want to backup. It can be as simple as a drive, as in C:; it can include a specific directory, as in C:\PROGS; or it can select specific files C:\PROGS*.EXE.
pathhname2	Selects the destination where you want the source files backed up. It can be as simple as a drive, as in C:; or it can include a specific directory, as in C:\PROGS.

Switches

/s	Backs up the subdirectories of the source path as well as the files in the source directory.
/m	Backs up files that have been modified since the last BACKUP command. Using this switch removes the archive attribute from any backed up files so, the next time this switch is used, only the files with an archive attribute are backed up.
/a	Adds the current backup to the end of another backup, so the new backup is tacked onto the end of the old one.

/f:*size*	Formats the disk before using it as a backup disk. The size must indicate whether you are using a 160K, 180K, 320K, 360K, 720K, 1.2 MB, 1.44 MB, or 2.88 MB disk. You can enter the number of kilobytes, number of kilobytes followed by a K or KB, the number of megabytes (1.2, 1.44 or 2.88) or the number of megabytes followed by an M or MB.
/d:*date*	Backs up files with a date on or after the specified date. The format of *date* must match the format of the date used by the DATE command.
/t:*time*	Backs up files with a time the same as or later than the specified time. The format of the time must match the format of *time* used by the TIME command.
/l:*pathname*	Creates a log file of the files the BACKUP command copies. If you do not provide a pathname, DOS creates a file called BACKUP.LOG on the root directory of the current disk.

How to use this command This command is frequently used to back up a hard drive onto several floppies, so that if something happens to the hard drive, you can restore the files from the backup disks. For example, you might enter the command:

```
BACKUP C: A: /S /M /L
```

This backs up all of the files on drive C, including the files stored in different subdirectories, since the /s switch is used. The /l switch tells DOS to create a file called C:\BACKUP.LOG that will list both the files it has backed up and the disk they are on. Since the /m switch is used, the archive attribute is removed from the backed up files. When you want to back up the newer files, you can enter the same command again and only the files that have changed since the last backup will be copied to the floppies.

Notes The files on the backup disks are not exactly the files on the original disk. The BACKUP command combines the files, so you must use the RESTORE command to restore one or more of the files you backed up. Also, since the backup disks are a continuous set of files, you want to keep all of the backup disks intact and together as a set. Make sure to label the disks so you can insert the proper one when you use the RESTORE command.

The BACKUP command returns the following exit codes, which you can use in a batch file with IF ERRORLEVEL:

0 Backup completed successfully.

1 No files were found to back up.

2 Some files were not backed up because you did not have access to them or someone else was using them.

3 You pressed CTRL-C to stop the BACKUP command.

4 BACKUP command halted due to an error.

Command type MS-DOS external command with network support.

BREAK

What this command does Controls when MS-DOS checks whether CTRL-C has been pressed. CTRL-C is used to halt a DOS command. DOS automatically checks for CTRL-C when it is sending information to or receiving information from the keyboard, printer or screen.

How to enter this command

BREAK [onloff]

How to customize this command

Parameters

No parameter entry	Displays the current BREAK setting.
onloff	Turns additional CTRL-C checking on or off. When additional CTRL-C checking is on, DOS also checks for CTRL-C when it reads or writes to the disk. The default is not to perform the additional CTRL-C checking.

How to use this command To have DOS check for CTRL-C when it reads and writes to a disk, enter the command BREAK ON to turn the additional checking on. Later you can enter BREAK OFF to return to the default of only checking when DOS performs reads or writes to the keyboard, printer or screen.

Notes You can also set DOS to perform additional CTRL-C checking in the CONFIG.SYS file by inserting the command:

BREAK=ON

Command type MS-DOS internal command that can be used in CONFIG.SYS.

BUFFERS

What this command does Allocates space for DOS to use as buffers. Buffers are used to temporarily store information that has been read from a file or will be written to a file. By increasing the number of buffers, you can increase the execution speed, but you also decrease the memory available for programs. You can also use a secondary cache to store additional information from files.

How to enter this command

BUFFERS=*m*[,*n*]

How to customize this command

Parameters

m	Specifies a number between 1 and 99 for the number of buffers. Each buffer uses 532 bytes.
n	Specifies the number of buffers in a secondary cache. The default is 1.

How to use this command If you wanted to increase the number of buffers to thirty, for example, you would insert this command in your CONFIG.SYS file:

```
BUFFERS=30
```

Notes If DOS is in the high memory, it will store the buffers in high memory. The default for the number of buffers is based on the RAM of the system. The defaults are: a 360 disk for less than 128K, 5 buffers for systems with between 128K and 255K of RAM, 10 buffers for systems with between 256K and 511K of RAM, and 15 buffers for systems with 512K of RAM or more. The optimal number of buffers depends on your hard disk size.

Command type Used solely in the CONFIG.SYS file.

CALL

What this Command Does Executes a second batch file within the execution of the first, and then performs the remaining commands in the first batch file.

How to enter this command

CALL [*drive:*][*path*]*filename* [*argument*]

How to customize this command

Parameters

drive	Selects the drive containing the batch file to execute if it is not on the current drive.
path	Selects the directory containing the batch file to execute if it is not in the current directory.
filename	Selects the batch file to execute. You do not need to provide a .BAT extension.
argument	Provides any arguments used by the called batch file.

How to use this command A batch file can use the CALL command in a statement like this:

```
IF %1==Y CALL MY_FILE
```

In this example, if the first variable equals Y, DOS performs the DOS commands stored in the file MY_FILE.BAT and then continues executing the commands in the original file. If the first variable does not equal Y, DOS ignores the CALL command and goes on performing the commands in the current file.

Command type Batch file command.

CHCP

What this command does Changes the current code page or displays the current code pages. You can change code pages to use a different character set that provides characters from a foreign language.

How to enter this command

```
CHCP [nnn]
```

How to customize this command

Parameters

No parameter entry	Displays active and prepared code pages.
nnn	The code page number to make active. These include: 437 (United States), 850 (Multilingual), 860 (Portuguese), 863 (Canadian-French), or 865 (Nordic).

How to use this command After using the COUNTRY and DEVICE commands in CONFIG.SYS, loading NLSFUNC, and loading the code page with MODE CP PREP, you can switch to another code page by entering:

```
CHCP 850
```

This switches the active code page to 850 (or whatever number you enter). Programs started after this command use the new code page and programs started before this command use the old code page.

Command type MS-DOS internal command with network support.

CHDIR or CD

What this command does Displays or changes the current directory.

How to enter this command

CHDIR [*pathname*]
CD [*pathname*]

How to customize this command

Parameters

No parameter entry	Displays the current path.
pathname	Selects the directory that you want to become the current directory. You can use (two dots) to make the parent directory (one level above the current directory in the directory tree structure) active.

How to use this command Use this command whenever you need to change directories. You might want to change the directory to execute a program stored in another directory or to perform actions on files in another directory without having to enter the full pathname each time. For example, to activate the \DBASE\CLIENT directory on the current drive you would type:

```
CD \DBASE\CLIENT
```

Command type MS-DOS internal command with network support.

CHKDSK

What this command does Displays a status report of a disk and conventional RAM, as well as checking the disk for logical errors in the file allocation and directories. Logical errors include: allocation units not used by files that are not included in the file allocation table as available for use, file sizes in the file allocation table that do not match the size of the file on the disk, and multiple files using the same allocation unit.

How to enter this command

CHKDSK [*pathname*] [/f] [/v]

How to customize this command

Parameter

pathname Selects the drive or file this command checks for logical errors and reports on the status. If you provide a drive, this command checks all files for logical errors and only reports the ones it finds. If you provide a file, or files, using the ? and * wildcard characters, this command reports logical errors for the files, and lists which files are fragmented by reporting the files that are stored in non-contiguous blocks and the number of non-contiguous blocks each one uses. (The files that are in non-contiguous blocks are fragmented since they are not stored in adjoining allocation units on the disk.)

Switches

/f Fixes reported logical errors.

/v Displays the names of the files as this command checks them for logical errors.

How to use this command This command is usually used to check that a disk contains no logical errors, as well as to display a status report on the disk. The status report contains the volume label, the date and time the disk was last formatted, its serial number, the disk size, how much disk space is used by your files, how much space is used by system files, how much space is used by hidden files, how much disk space is used by bad sectors, the allocation unit size, the number of allocation units on the disk, the available allocation units for new files, the amount of conventional RAM, and the available conventional RAM. To see this information for the disk in drive A, for example, enter

CHKDSK A:

If the disk contains logical errors, DOS will ask you if you want to convert them into chains. If you include the /f switch, you can type **Y** to convert the lost chains into files with a filename of FILE0000.CHK (or a higher number).

Notes Do not use CHKDSK with a disk that contains open files (such as when you are in the middle of a program and you exit to DOS, or while multi-tasking). Do not use CHKDSK on drives created with the SUBST, JOIN, or ASSIGN commands.

Command type MS-DOS external command.

CLS

What this command does Clears all the information from the screen when you are working at the command prompt.

How to enter this command

```
CLS
```

How to customize this command No parameters or switches are used with this command.

How to use this command You might type CLS and press ENTER to eliminate the error messages presented from several erroneous attempts at entering a DOS command.

You can use CLS in a batch file that presents a menu. This command will ensure that the menu begins at the top of your screen.

Notes If you have enabled DOSKEY, the DOS commands you enter at the DOS prompt will be recorded for later use. CLS does not clear this history, so the commands will still be available even though they do not appear on your screen.

Command type MS-DOS internal command with network support.

COMP

What this command does Compares two files and reports the differences. The COMP command is better than DISKCOMP to compare files on two disks when the files were not copied to the disk with the DISKCOPY command. This is because the files may be stored in different locations on each disk, which DISKCOMP will report but COMP will not, assuming the contents of the files are identical. The

default output the *COMP* command produces displays the offset (number of bytes from the beginning where there is a difference) and the hexadecimal value of the two different characters. This command stops comparing two files when it finds more than ten differences.

How to enter this command

COMP [*pathname1*] [*pathname2*] [/a] [/d] [/l] [/n:*number*] [/c]

How to customize this command

Parameters

pathname1 Selects the first file that you want to compare. You can use the wildcards * and ? to specify the filenames, or you can provide only the path information to compare all of the files in the path.

pathname2 Selects the second file that you want to compare. You can use the wildcards * and ? to specify the filenames, or you can provide only the path information to compare files of the same name among the files that are specified with *pathname1*. If you do not provide *pathname1* and *pathname2*, DOS prompts you for the pathnames to compare and for the options you want to use. You can also use the same entries for *pathname1* and *pathname2*, and DOS will prompt you when you should insert each disk.

Switches

/a Displays the differences between the files in ASCII format, which shows the bytes that are different as the characters they represent.

/d Displays the differences between the files in decimal format which shows the bytes as their decimal ASCII value.

/c Causes COMP to ignore differences that are caused by the same letter being in different cases.

/l Displays the line number containing the difference, rather than the offset.

/n:*number* Sets the number of lines from each file, starting at the beginning, that COMP compares. This switch only compares part of a file rather than comparing the entire file. Number must be the number of lines you want the command to compare.

How to use this command The COMP command does a binary comparison of two files that does not resynchronize files the way the FC command does. It is useful when you want to compare non-text files, such as checking that the .EXE files on a floppy match the .EXE files on your hard disk. To compare two files such as

PROGRAM.EXE on the current drive and the same file on drive A, for example, you would enter:

```
COMP PROGRAM.EXE A:
```

DOS will compare the two files. After reporting either that the files are identical or that they have up to ten differences, DOS will compare a new set of files if you type **Y** and then enter a new entry for *pathname1* and *pathname2* at the prompts.

Command type MS-DOS external command with network support.

COPY

What this command does Copies files to another location, or combines several files into one.

How to enter this command

```
COPY [pathname1] [pathname2] [/v] [/a] [/b]
COPY [pathname1] [/v] [/a] [/b] [pathname2]
COPY pathname1+pathname2[+...] pathnameN
```

How to customize this command

Parameters

pathname1 Selects the file or files to copy to a new location. Unless you include different information, DOS assumes the files are in the current directory on the current drive.

pathname2 When used to copy files, selects the destination of the files that are copied. It can include a drive, path information, or even new file names if you want the new copy to use different filenames. If you omit this information, DOS tries copying the files to the current drive and directory using the same filenames. When *pathname2* is used to append files, it is joined to pathname1 with a plus sign as the other files to be combined are joined with plus signs.

pathnameN Selects the destination file that will contain the combined files. If you omit this information DOS tries copying the files to the current drive and directory using the first filename provided.

Switches

/v	Checks each file after it is copied.
/a	Copies files until the end-of-file character (CTRL-Z) when this switch precedes *pathname2*. When this switch follows *pathname2*, DOS adds an end-of-file character. This is the default when you use the COPY command to combine files.
/b	Copies files until the number of bytes determined by a file's size is copied. When this switch follows *pathname2*, DOS does not add an end-of-file character. This is the default when you use the COPY command without combining files.

How to use this command You can use this command to copy files, send files to devices and combine files. To copy a file such as MY_DATA.TXT to another drive, for example, you would enter:

```
COPY MY_DATA.TXT A:
```

This copies the file MY_DATA.TXT in the current directory to drive A. Since a filename was not included after A:, DOS uses the same filename for the copy. When you copy files, all of their attributes are copied as well. You can also use wildcards to select groups of files, as in:

```
COPY C*.* A:D*.*
```

This example copies all files in the current directory that start with the letter C to drive A and changes the C at the beginning of each filename to a D. Another possibility for the COPY command is to use a device as either the source or the destination. For the source, you may want to use CON so you can enter text directly into a file:

The ^Z ends the file when you press ENTER. You can enter ^Z by pressing either CTRL-Z or F6. Another example of using a device with the COPY command as the destination is:

This copies the contents of MY_FILE.TXT to the printer connected to the first parallel port.

The last use of the COPY command is combine files, as in:

```
COPY MY_FILE.TXT+MY_FILE.DAT+DATAFILE ALL_FILE
```

This example joins the contents of MY_FILE.TXT, MY_FILE.DAT, and DATAFILE, and stores the combined file in the ALL_FILE file. It is possible to use this command accidentally. For example, if you type COPY *.* A to copy files to drive A, forgetting the colon, all of the files in the current directory will be copied to a file called A.

Command type MS-DOS internal command with network support.

COUNTRY

What this command does Sets the country conventions used by MS-DOS for such things as date and time formats, case conversions, currency, sorting, folding formats and decimal separators. If this command is not used, MS-DOS uses the United States as the default. Since you select a country during MS-DOS 5 installation, this command should have been entered for you. You would only need to use it again if you want to change the country DOS uses from the original setting.

How to enter this command

COUNTRY=*xxx*[,*yyy*][,*pathname*]

How to customize this command

Parameters

xxx	Specifies the country code for the country to use.
yyy	Specifies the code page to use with the country settings.
Pathname	Specifies the file containing the information for the different countries. This is normally C:\DOS\COUNTRY.SYS. If you do not provide it, DOS tries using COUNTRY.SYS in the startup directory.

How to use this command If you wanted to change the country code to another country, such as Germany, you would enter this command in your CONFIG.SYS file:

```
COUNTRY=049,,C:\DOS\COUNTRY.SYS
```

After you entered this command, the date would appear as 30.12.1990 for December 30, 1990 and the time would appear as 16.37.00,00 for 4:37 in the afternoon.

Notes The country code affects the DATE, BACKUP, RESTORE and TIME commands. DATE and TIME will require new entries to use the date and time formats of the currently selected country code. BACKUP and RESTORE use the date and time to determine which files are backed up and restored.

Command type Used solely in the CONFIG.SYS file.

CTTY

What this command does Redirects the default input and output from the keyboard and screen to another device.

How to enter this command

CTTY *device*

How to customize this command

Parameter

device Selects the device that the computer will use for default input and output. This is usually COM1 or AUX.

How to use this command You can use this command to change the device that DOS accepts input from and sends output to, from the keyboard and screen to another device. For example, you might enter:

```
CTTY AUX
```

When you want to return to using the keyboard and screen, enter:

```
CTTY CON
```

Notes If you use CTTY to redirect the input and output to a serial port, you must first use the MODE command to make any changes to the serial port, then use the CTTY command.

Command type MS-DOS internal command with network support.

DATE

What this command does Displays or sets the current date in your computer system. You can accept the displayed date by pressing ENTER, or you can type a new date to change the date in your computer. DOS will maintain this new setting during the current session. In some newer model computers, this change will remain in effect during future sessions.

How to enter this command

DATE [*date*]

How to customize this command

No parameter entry	Displays the current date and prompts for the new date. You can accept this date by pressing ENTER or enter a new date using a mm-dd-yy or mm/dd/yy format.
date	Sets the system date to the date you supply, assuming the date is a proper format as described above.

How to use this command If you are using an older model PC when you enter DATE you will see the following kind of response:

```
Current date is Tue 01-01-1980
Enter new date (mm-dd-yy):
```

If the current date were September 20, 1991, for example, you would type 9-20-91 and press ENTER.

Notes Each component of your date response must be valid. For example, you cannot enter a date of 2-30-90, since February never has 30 days. You do not have to enter leading zeros for the date components. If you change the country conventions with the COUNTRY command in CONFIG.SYS, the format of the date display and entry will change to match the country selected.

Command type MS-DOS internal command with network support.

DEBUG

What this command does Starts the DEBUG program, which lets you test and monitor executable files. This program is designed to help you find errors in other programs. Since the DEBUG program works in assembler code, you should only use this program if you are familiar with assembler code.

How to enter this command

DEBUG [*pathname* [*arglist*]]

How to customize this command

Parameters

pathname	Selects the executable file to load into DEBUG that you want to monitor and test as it runs.
arglist	Provides the arguments that the executable file specified by *pathname* requires to run.

How to use this command If you wanted to monitor how DOS performs an executable file such as MY_FILE.EXE, you could, for example, enter the following command:

```
DEBUG MY_FILE
```

Once DEBUG is loaded, you will see the DEBUG prompt, which is a hyphen. The DEBUG command has its own commands which are beyond the scope of this book. You can leave DEBUG by typing **Q** and pressing ENTER.

Command type MS-DOS external command.

DEL or ERASE

What this command does Removes one or more files from a disk. DOS deletes files by changing the file entry on the file allocation table so that the disk space file used is marked as available. DOS will then use that disk area to store other files.

How to enter this command

DEL *pathname* [/p]
ERASE *pathname* [/p]

How to customize this command

Parameters

pathname Selects the file to delete. You can use the * and ? wildcard characters to select groups of files to delete. If you do not provide a file extension, DOS only deletes matching files that do not have a file extension (unlike the DIR command).

Switch

/p Prompts for confirmation to delete each file that you select with *pathname*.

How to use this command This command is used to delete one or many files. You can use this command to delete only one file, as in:

```
DELETE BUDGET.DAT
```

You can also use it to delete a group — or even all — of the files, as in:

```
DEL  A:\*.* /p
```

The use of the /p switch prompts for confirmation that you want to delete the files. You must type **Y** to delete each file. Using a prompt lets you select which files in the group you will delete. Also, deleting all of the files in the directory displays an additional prompt of confirmation that you intend to delete all of the files in the directory.

Notes You cannot delete hidden, system or read-only files.

Command type MS-DOS internal command with network support.

DEVICE

What this command does Loads a device driver into memory when MS-DOS reads the CONFIG.SYS file. Device drivers offer additional features, including on-screen graphics, changing code pages, creating virtual drives, disk caching, and using additional hardware such as a mouse.

How to enter this command

DEVICE=[*drive:*][*path*]*filename* [*arguments*]

How to customize this command

Parameters

drive	Specifies the drive containing the device driver to load, if it is not the current one.
path	Specifies the path containing the device driver to load, if it is not the current one.
filename	Specifies the name of the device driver.
arguments	Provides the arguments used by the device driver. These are unique for each device driver.

How to use this command If you wanted to load the device driver RAMDRIVE to create a virtual drive, and wanted this drive to be 128K, you would include this command in the CONFIG.SYS file:

```
DEVICE=C:\DOS\RAMDRIVE.SYS 128
```

The 128 is an argument that the RAMDRIVE.SYS device driver uses. If you used another device driver, you would probably include other information after the device driver filename.

Command type Used solely in the CONFIG.SYS file.

DEVICEHIGH

What this command does Loads a device driver into reserved memory when MS-DOS reads the CONFIG.SYS file. Device drivers offer additional features including on-screen graphics, changing code pages, creating virtual drives, disk caching, and using additional hardware such as a mouse. Not all device drivers can be loaded into reserved memory.

How to enter this command

DEVICEHIGH=[*drive*:][*path*]*filename* [*arguments*]
DEVICEHIGH SIZE=*hexsize* [*drive*:][*path*]*filename* [*arguments*]

How to customize this command

Parameters

hexsize	Specifies the number of bytes, in hexadecimal form, that must be available in reserved memory before DOS loads the device driver into reserved memory. If available reserved memory is too small, DOS loads the device driver in conventional memory.
drive	Specifies the drive containing the device driver to load, if not the current one.
path	Specifies the path containing the device driver to load, if not the current one.
filename	Specifies the name of the device driver.
arguments	Provides the arguments used by the device driver. These are unique for each device driver.

How to use this command If you wanted to load the device driver RAMDRIVE to create a virtual drive in reserved memory and wanted this drive to be 128K, you would include this command in the CONFIG.SYS file:

```
DEVICEHIGH SIZE=28000 C:\DOS\RAMDRIVE.SYS 128
```

The SIZE argument prevents DOS from loading a device driver into reserved memory when there is not enough room. The 128 is an argument that the RAMDRIVE.SYS device driver uses. If you used another device driver, you would probably include other information after the device driver filename.

Notes Before you use the DEVICEHIGH command, your CONFIG.SYS file must include the DOS=UMB and DEVICE=EMM386.EXE commands.

Command type Used solely in the CONFIG.SYS file.

DIR

What this command does Lists one or more files in a directory.

How to enter this command

DIR [*drive:*][*path*][*filename*] [/p] [/w] [/a:*attributes*] [/o:*sortorder*] [/s] [/b] [/l]

How to customize this command

Parameters

drive	Selects the drive that you want to list from if it is not the current drive.
path	Selects the directory containing the files to list, if it is not the current directory.
filename	Selects the files to list. You can include the * and ? wildcard characters. If you do not provide a filename, DOS assumes *.* (all files in the directory). If you do not provide a file extension (which is different than typing a period at the end to indicate that there is no extension), DOS assumes *.

Switches

/p	Displays one screen of files at a time and pauses. You must press a key to see the next screen.
/w	Displays a wide display of the filenames which displays only the filenames using five columns across the screen.
/a:*attributes*	Displays only files that match the selected attribute. Attributes must be: blank for all files including hidden and system files, h for hidden files, -h for non-hidden files, s for system files, -s for non-system files, d for directories, -d for files without directories, a for files with the archive attribute, -a for files without the archive attribute, r for read-only files or -r for files that are not read-only. You can combine these attribute letters, as in /a:hr to list only read-only files and hidden files. Only use the colon if you are putting one of these characters after it. If you do not use this switch, DOS lists only files and directories that are not hidden or system files.
/o:*sortorder*	Displays the file listing in a sorted order. Sortorder must be: blank to sort directories before the files in alphabetical order, n for alphabetical filenames, e for alphabetical by extension, d for in order by date and time, s for ascending size order, or g to group directories before files. You can add a hyphen before the n, e, d, s, or g to reverse the order of the listing. You can combine sort orders, as in /o:ed, which sorts the files in order by their extension and then sorts those with the same extension by their date. Only use the colon if you are putting one of these characters after it. If you do not use this switch, DOS lists files and directories in the order they are listed in the file allocation table.
/s	Lists the files in the selected path's subdirectories as well as the files in the selected directory.

/b	Displays only filenames and extensions with each file on a separate line. If /b is combined with the /s switch, DIR lists the matching files and includes the path information before the filename.
/l	Changes the display from uppercase to lowercase.

How to use this command If you wanted to list the files starting with WEEK on drive A in the DATES subdirectory in alphabetical order, you would make the following entry:

```
DIR A:\DATES\WEEK*  /O:N
```

DOS will display the drive serial number; volume name; matching files, including the date and time they were saved, and their size; the number of bytes the listed files use; and the number of bytes remaining on the disk.

Notes You can create default DIR command settings by setting the environment variable DIRCMD to the options you want. For example, you can enter the command SET DIRCMD=/O:N /P to have all subsequent DIR commands use these two switches. To override a default DIR command switch, you must include the switch with a hyphen between the slash and the letter, as in DIR /-O /-P.

Command type MS-DOS internal command with network support.

DISKCOMP

What this command does Compares two disks to see if they are the same except for different serial numbers.

How to enter this command

DISKCOMP [*drive1*:] [*drive2*:] [/1] [/v]

How to customize this command

Parameters

drive1	Selects the first disk to compare.

drive2 Selects the second disk to compare. If this is the same disk as *drive1*, DOS will prompt you when you should insert the first disk and when you should enter the second disk. If *drive2* is not provided, DOS assumes it is the same as *drive1*. If neither *drive1* nor *drive2* is provided, DOS assumes that the current drive is both drives. If the current drive is a hard disk, this command stops without comparing disks.

Switches

/1 Compares only the first side of the disk, which 160K and 180K disks use.

/8 Compares only the first eight sectors of the disk, which 160K and 320K disks use.

How to use this command This command is used to tell whether that two disks contain the same information. You can use this, for example, to check that a disk you created with DISKCOPY still contains the same information as the original. To compare a disk in drive A with a disk in drive B, for example, you would enter:

```
DISKCOMP A: B:
```

DOS will display a prompt to put the first disk in drive A and the second disk in drive B. When you press a key, DOS will compare the two disks. After the disks are compared, you can compare a new set of disks by typing **Y** and pressing a key after inserting the next set of diskettes.

Notes If you use this command on the default drive from which you booted the computer, you may need to put the boot disk in drive A, or another disk with COMMAND.COM in drive A, and press any key when this command is finished. This command returns the following exit codes, which you can use in a batch file with IF ERRORLEVEL:

0	The two disks are identical
1	The two disks are not the same
2	You pressed CTRL-C to stop the command
3	Fatal hard error
4	Initialization error

Command type MS-DOS external command.

DISKCOPY

What this command does Duplicates a disk. The copy disk must be the same capacity as the original disk. The copy disk will automatically be formatted if required.

How to enter this command

DISKCOPY [*drive1:*] [*drive2:*] [/1] [/v]

How to customize this command

Parameters

drive1	Selects the disk that you want to copy, also called the source disk.
drive2	Selects the disk that will receive the copy, also called the target disk. If this is the same disk as *drive1*, DOS will prompt you when you should insert the source disk and when you should enter the target disk. If *drive1* s not provided, DOS assumes it is the current disk. If neither *drive1* or *drive2* is provided, DOS assumes that the current drive is both the source and target drives. If the current drive is a hard disk, this command stops without copying data.

Switches

/1	Copies only the first side of the disk, which 160K and 180K disks use.
/v	Verifies that the disk is copied correctly.

How to use this command This command is used to quickly make backup copies of your floppies and archival copies of purchased software. For example, to make a backup copy of a disk in drive A on another disk in drive B, enter:

```
DISKCOPY A: B: /V
```

DOS will display a prompt to put the source disk in drive A and the target disk in drive B. When you press any key, DOS will copy the diskette in drive A, formatting the disk in drive B, if necessary, to the same capacity as the disk in drive A. Any data previously on drive B will be erased. This command makes an exact copy of the original, except the serial numbers of the two disks will be different. After the disks are copied, you can copy a new set of disks by typing **Y** and pressing any key after inserting the next set of diskettes.

Notes This command returns the following exit codes, which you can use in a batch file with IF ERRORLEVEL:

0	The disk was copied successfully.

1 Nonfatal read/write error (such as using this command with a hard disk)
2 You pressed CTRL-C to stop the command
3 Fatal hard error
4 Initialization error

Command type MS-DOS external command.

DOS

What this command does Selects where DOS is loaded and whether the link to reserved memory is preserved after the booting process so you can use the available reserved memory for other applications and device drivers.

How to enter this command

```
DOS=highllow
DOS=umblnoumb
```

How to customize this command

Parameters

highllow	Determines where DOS is loaded. High loads most of DOS in the high memory area, with a small part in conventional memory. Low, the default, loads all of DOS in conventional memory.
umblnoumb	Selects whether the link to reserved memory is maintained after the device drivers that DOS loads during startup are loaded. Umb retains the link, and noumb, the default, drops this link.

How to use this command Use this command to take advantage of the extra memory so you have more conventional memory for other applications. You might include a command like this, for example, in your CONFIG.SYS file:

```
DOS=HIGH,UMB
```

This command loads DOS in reserved memory and retains the link to reserved memory so you can load other device drivers there using DEVICEHIGH, or other programs using LOADHIGH. This command would be placed in the CONFIG.SYS file after the DEVICE=HIMEM.SYS and DEVICE=EMM386.EXE commands. As this

example shows, you can combine the two parameters by separating them with a comma.

Command type Used solely in CONFIG.SYS.

DOSKEY

What this command does Remembers executed DOS commands that you can repeat, edit and use to create macros. While DOSKEY is loaded, you can use several keys to switch to a different command that DOSKEY remembers, or use several keys to alter the command displayed on the command prompt. The DOSKEY program is described in greater detail in Chapter 10.

How to enter this command

DOSKEY [/reinstall] [/bufsize=*size*] [/macros] [/history] [/insertloverstrike] [macro=*text*]

How to customize this command

Parameters

No parameter entry	Loads the DOSKEY program using the default settings.
macro=*text*	Creates a DOSKEY macro. Macro must be the macro name and text must be the commands that the macro performs. If you want to delete a macro, do not supply any text after the equals sign.

Switches

/reinstall	Reinstalls the DOSKEY program over an existing one so you can load DOSKEY with different settings.
/bufsize=*size*	Selects the size of the buffer that DOSKEY uses to store DOS commands and keystrokes. Size is the number of bytes. If you do not use this switch, DOS uses the default of 512 bytes.
/macros or /m	Displays a list of DOSKEY macros. By combining this switch with a redirection symbol such as >, you can store these macros in a file.
/history or /h	Displays a list of DOS commands stored in the buffer DOSKEY uses. By combining this switch with a redirection symbol such as >, you can store these commands in a batch file.

/insertloverstrike — Selects whether a new character is inserted into the DOS command line (/insert) or replaces the current character (/overstrike). The default is /overstrike. You can switch between the two while editing a command by pressing the INSERT key, but DOS returns to insert or overstrike as set by this switch when you press ENTER.

How to use this command If you want to load the DOSKEY command so you can use it to recall the DOS commands you enter, make the following entry:

```
DOSKEY
```

Once it is loaded, you can use the UP ARROW and DOWN ARROW to switch to a previously executed command. You can modify previously executed commands as they are displayed.

Notes This command remains in memory, so you will have about 3K less memory available for other commands.

Command type MS-DOS external command with network support.

DOSSHELL

What this command does Loads the Shell program, which provides a user-friendly interface to DOS. The Shell is covered in more detail in Chapters 4 and 5.

How to enter this command

```
DOSSHELL [/t:llm|h|v|s|u] [/g:llm|h|v|s|u] [/b] [/n] [/s]
```

How to customize this command

Switches

/t:llm|h|v|s|u — Loads the Shell using a text screen of: low, medium, high, very high, superhigh or ultrahigh quality. Place the letter of the quality you want after the colon. You can also use the default text screen quality by entering only /t. The capability of the screen depends on your monitor. You cannot use this switch with the /g switch.

/g:llm|h|v|s|u — Loads the Shell using a graphics screen of: low, medium, high, very high, superhigh or ultrahigh quality. Place the letter of the quality you want after the colon. You can also use the default screen quality by entering only /g. The capability of the screen depends on your monitor. You cannot use this switch with the /t switch.

/b Sets the DOS Shell to black and white.

How to use this command If you wanted to start the DOS Shell and set the graphics screen quality to high, for example, you would make the following entry:

```
DOSSHELL /g:h
```

Command type MS-DOS external command.

ECHO

What this command does Displays a prompt message during the execution of a batch file, displays the current status of ECHO, or turns the ECHO on or off. It can be used to display an entire screen of text that might offer the user a menu of choices. When "ECHO" is on, every command is displayed as it is executed. When ECHO is off, commands are processed without displaying.

How to enter this command

```
ECHO[onloff] [message]
```

How to customize this command

Parameters

onloff	Turns the command echoing on or off.
message	Specifies the text to display on the screen.

How to use this command If you wanted to hide the contents of a batch file, you might include these statements at the beginning of the file:

```
ECHO OFF
ECHO Now the batch file will process the data.
```

The first command turns off command echoing and the second displays a message on the screen.

Notes You can prevent specific lines in a batch file from echoing by including an at sign (@) before the command.

Command type Batch file command.

DRIVPARM

What this command does Sets the parameters used by block devices such as disk drives.

How to enter this command

DRIVPARM=/d:*number*[/c][/f:*factor*][/h:*heads*][/i][/n][/s:*sectors*][/t:*tracks*]

How to customize this command

Switches

/d:*number*	Chooses the physical drive number of the block device you are setting the parameters for.
/c	Indicates that the disk drive can tell if the door is not closed. If the disk drive can, MS-DOS will know when you have put a new disk in the drive it is waiting for.
/f:*factor*	Selects the drive type The choices for factor are:
0	for 360K or less disk drives
1	for 1.2 MB disk drives
2	for 720K 3-1/2 inch disk drives
5	for hard disk drives
6	for tape disk drives
7	for 1.44 MB 3-1/2 inch disk drives
	If you do not supply it, DOS assumes 2 for a 720K 3-1/2 inch disk drives.
/h:*heads*	Selects the number of drive heads for the block device. The default depends on the entry for the /f:factor.
/i	Indicates that you are using a 3-1/2 inch disk drive on an IBM PC/AT compatible which does not have the internal software (that is, built into the machine) to support 3-1/2 inch disk drives.
/n	Indicates that the block drive is non-removable.
/s:*sectors*	Selects the number of sectors on each track. The default depends on the entry for the /f:switch.
/t:*tracks*	Selects the number of tracks on each side of the block device. The default depends on the entry for the /f:switch.

How to use this command If you wanted to set drive E to a non-removable hard disk drive that uses 32 sectors, you would enter this in the CONFIG.SYS file:

```
DRIVPARM=/D:4 /F:5 /N /S:32
```

Command type This command is used solely in CONFIG.SYS.

EDIT

What this command does Loads the DOS Editor, which you can use to edit ASCII text files. These files include batch files, ASCII files created by other programs, and files created by redirecting the output of other DOS commands. Menu options let you retrieve and save files, print them, and search them for text. The Editor menu is further covered in Chapter 11.

How to enter this command

```
EDIT [drive:][path][filename] [/h] [/nohi] [/b] [/g]
```

How to customize this command

Parameters

drive	Selects the drive containing the file that you want to edit, if it is not on the current drive.
path	Selects the directory containing the file that you want to edit, if it is not on the current directory.
filename	Selects the file that you want to edit.

Switches

/h	Displays the maximum number of lines on the screen that your hardware is capable of displaying. This is usually 25 lines for CGA screens, 43 lines for EGA screens, and 50 for VGA screens.
/nohi	Sets the editor to not use high intensity, so you can see the screen on portable computers and some monochrome screens.
/b	Displays the editor in black and white instead of color.
/g	Updates CGA screens faster to eliminate the "snowy" appearance.

How to use this command If you wanted to edit a file called MY_FILE.BAT using the DOS Editor, you would enter the following:

```
EDIT MY_FILE.BAT
```

When you want to leave EDIT, press ALT and select File and Exit. With a mouse, click on File in the menu bar and Exit in the File menu.

Command type MS-DOS external command.

EDLIN

What this command does Loads the EDLIN editor which is a line editor you can use to edit ASCII text files. EDLIN does not have the menu options available with the EDIT DOS Editor. The EDLIN commands are listed in Chapter 11.

How to enter this command

EDLIN [*drive:*][*path*]*filename* [/b]

How to customize this command

Parameters

drive	Selects the drive containing the file that you want to edit, if it is not on the current drive.
path	Selects the directory containing the file that you want to edit, if it is not on the current directory.
filename	Selects the file that you want to edit.

Switch

/b	Causes EDLIN to ignore CTRL-Z characters, so it reads in the number of characters selected by the file size in the file allocation table.

How to use this command If you wanted to edit a file called MY_FILE.BAT using EDLIN, you would enter the following:

```
EDLIN MY_FILE.BAT
```

When you want to leave EDLIN, type **q** and press ENTER.

Command type MS-DOS external command.

EMM386

What this command does Enables or disables EMM386 expanded memory.

How to enter this command

EMM386 [on|off|auto] [w=on|w=off]

How to customize this command

Parameters

on|off|auto	Controls use of the EMM386.EXE device driver. The three choices activate the device driver, suspend it, or place it in auto mode, which enables Weitek Coprocessor support when a program asks for it.
w=on|w=off	Enables or disables Weitek Coprocessor support.

How to use this command If you want to use EMM386 expanded memory, you can use this command:

EMM386 ON

Command type MS-DOS external command.

Notes To use this command at the DOS command prompt or in a batch file, you must include a DEVICE-EMM386.EXE command in CONFIG.SYS. This command is only available to computers with a 80386 or 80486 processor.

EXE2BIN

What this command does Converts executable files with a .EXE extension to binary files with a .BIN extension. This is primarily intended for programmers, so you should only use this command if you are writing your own program or have instructions to use this command from a software developer.

How to enter this command

EXE2BIN *pathname1* [*pathname2*]

How to customize this command

Parameters

pathname1	Selects the executable file to convert. You do not need an extension if it is .EXE.

pathname2 Selects the name of the resulting binary file. If you do not provide this information, the binary file will be in the current directory, using the same filename as the executable file but, with a .BIN extension.

How to use this command If you had created a program called MY_FILE.EXE that you wanted to convert to a binary file, you would enter:

```
EXE2BIN MY_FILE
```

Notes The executable file must be in a .EXE file format, which is produced by a linker. The resident and data code section of the .EXE file must be less than 64K. The executable file must not have a STACK segment.

Command type MS-DOS internal command with network support.

EXIT

What this command does Exits the current COMMAND.COM command processor or another command processor and returns to the next previous command processor. This command is only used when multiple processors are loaded. This command also returns you to applications, such as WordPerfect or 1-2-3, that provide a temporary exit to DOS. If DOS cannot leave the current command processor, the EXIT command is ignored.

How to enter this command

```
EXIT
```

How to use this command When you have started another command processor, or are using software that provides a temporary exit to DOS, you can return to the prior command processor or that software by typing EXIT and pressing ENTER.

Command type MS-DOS internal command with network support.

FASTOPEN

What this command does Tracks the location of files and directories, as well as caching the file data, so data access is quicker. When you open a file, FASTOPEN records the location of the file so the next time the file is needed, DOS knows where it is. This command is usually executed from the CONFIG.SYS file or the AUTOEXEC.BAT file.

How to enter this command

FASTOPEN *drive*:[=*n*] [/x]

or:

INSTALL=FASTOPEN.EXE *drive*:[=*n*] [/x]

How to customize this command

Parameter

drive=n Selects the drive on which you want to track the files that are opened and the directories they use. N, a number between 10 and 999, sets the number of files that this command tracks. If you do not use =n, FASTOPEN uses 10. This parameter can be repeated for every drive that you want to use with FASTOPEN.

Switch

/x Stores the cache this command creates in expanded memory.

How to use this command To improve disk performance for drives C and D and use 150 files on each disk, for example, you would enter:

```
FASTOPEN C:=150 D:=150
```

Notes This command does not work on floppy disks and network drives. The settings made with FASTOPEN remain in effect during the current DOS session.

FASTOPEN remains in memory, which reduces the memory available for other programs by about 5K plus about 48 bytes for each file FASTOPEN tracks (the value of the n parameter).

Command type MS-DOS external command.

FC

What this command does Compares two files and reports the differences. Unlike the COMP command, the FC command tries to resynchronize the files once it finds differences. Also, the output of the FC command is easier to understand when you are comparing text files. The FC command is better than DISKCOMP to compare files on two disks when the files were copied to the disk with a command other than DISKCOPY. This is because the files may be stored in different locations on

each disk, which DISKCOMP will report but FC will not, assuming the contents of the files are identical.

How to enter this command

FC [/a] [/b] [/c] [/l] [/lb*n*] [/n] [/t] [/w] [/*nnnn*] [*pathname1*] [*pathname2*]

How to customize this command

Parameters

pathname1 Selects the first file that you want to compare. You can use the * and ? wildcards to specify the filenames, but you must provide a filename as well as a path.

pathname2 Selects the second file that you want to compare. You can use the * and ? wildcards to specify the filenames, but you must provide a filename as well as a path.

Switches

/a Abbreviates the display of the differences between the two files, showing only the beginnings of the lines that are different, followed by ellipses (...).

/b Performs a binary comparison much like the COMP command performs. This switch cannot be combined with the other FC switches. This is the default for files with .EXE, .COM, .SYS, .OBJ, .LIB, or .BIN extensions.

/c Ignores differences that are caused by letters being in different cases.

/l Performs an ASCII comparison, which produces a different display than the binary comparison. This is only used when you are comparing files that have .EXE, .COM, .SYS, .OBJ, .LIB, or .BIN extensions but you do not want a binary comparison.

/lb*n* Sets the number of lines from each file that DOS stores in the internal buffer during execution of the FC command. N is the number of lines you want. The number of lines in the buffer sets the number of lines DOS remembers as it tries to resynchronize two files after finding a difference. If the amount of memory is less than the FC command needs, it will store less than 100 lines (the default) or the number of lines set with this switch.

/n Displays the line numbers of the lines containing differences.

/t Prevents DOS from expanding tabs into eight spaces.

/w Compresses tabs and multiple spaces into a single space, so differences caused by different numbers of spaces or tabs are not reported by the FC command.

/*nnnn* Sets the number of lines that must be identical after a difference before the FC command recognizes that the files are resynchronized. nnnn is the number of lines.

How to use this command This command is especially useful for comparing two text files. The output this command produces is more legible for text files than the output of the COMP command. To compare two files such as BUDGET on drive A and the same file on drive B, you would enter:

```
FC A:BUDGET B:BUDGET
```

For each difference, FC reports: the filename of the first file, the lines that are different, the first line that matches, the filename of the second file, the lines that are different, and the first line that matches as the FC command resynchronizes the two files.

Command type MS-DOS external command with network support.

FCBS

What this command does Sets the number of file control blocks that programs can open concurrently. File control blocks are used in some older software.

How to enter this command

```
FCBS=x
```

How to customize this command

Parameter

x	Selects the number of file control blocks that you want to open concurrently. This must be a number between 1 and 255. If you don't use this command, DOS assumes the default of 4.

How to use this command If you are using old software that works with as many as eight files using file control blocks, you will need to increase the number of file control blocks by including this line in your CONFIG.SYS file:

```
FCBS=8
```

Notes Only use this command to change the number of file control blocks if you are using software that uses file control blocks to manipulate files, and you know that the number needs to be increased.

Command type This command is used solely in the CONFIG.SYS file.

FDISK

What this command does Displays or changes how a hard disk is divided into partitions or how an extended DOS partition is divided into logical drives.

How to enter this command

FDISK

How to use this command FDISK offers you the use of menus rather than using parameters and switches. From the initial menu, you can select: add a partition or logical drive, change the active partition, delete a partition or logical drive, display the current partition information, or change the hard disk that the FDISK command uses (if you have more than one physical hard disk).

When you divide a hard disk into partitions or logical drives, subsequent DOS commands treat them as separate hard drives. A hard disk can contain one primary DOS partition (the one containing the DOS files used to boot the system), extended DOS partitions, and non-DOS partitions (ones that use different operating systems) as long as the total number of partitions on each hard disk is four or fewer.

When you add a partition or logical drive, you select the number of megabytes or percentage of the hard disk to be used by the partition or logical drive. Each primary partition and logical drive (or extended drive if the extended drive is not divided into partitions) is assigned a letter, starting with C if the computer has one or two floppy drives. The active partition is the partition the computer uses to boot from. When you leave the FDISK menus, if you have changed the partition information or changed which partition is active, when you press ENTER the system will restart.

Notes Changing the partition information makes the information that was previously stored on the disk unavailable. The only way you can later access the files on the hard disk where you have changed the partition is by using the UNFORMAT command with the /partn switch, assuming you have used the MIRROR command earlier with the /partn switch.

Do not use FDISK with drives that are created with the SUBST or JOIN commands. DOS can now handle large capacity disks, so you are not required to divide high capacity disks into partitions as prior releases required.

Command type MS-DOS external command.

FILES

What this command does Sets the number of files that you can open at one time using system calls. This command allocates space for DOS to use as a table for all of the open files. Most software works with several files at once, so the number of files that can be open must be equal to or greater than the number of files you expect to use in an application.

How to enter this command

FILES=*x*

How to customize this command

Parameter

x Specifies a number between 8 (the default) and 255 for the number of files you want to be able to open at one time.

How to use this command If you wanted to increase the number of files to thirty, for example, you would enter this command in your CONFIG.SYS file:

```
FILES=30
```

Command type Used solely in the CONFIG.SYS file.

FIND

What this command does Locates text in a file.

How to enter this command

FIND [/v] [/c] [/n] [/i] "*string*" *pathname*

How to customize this command

Parameters

string Supplies the text this command will FIND.

pathname Supplies the name of the file or files to search. You cannot use the * or ? wildcards to select multiple files, but you can list more than one file to search by separating them with a space. If you do not supply a pathname, DOS uses this command as a filter, which expects to receive input using the >, < or | redirection characters.

Switches

/v	Displays the lines that do not contain the string.
/c	Displays the number of lines in each file that contain the string. You cannot use this switch with /n. If you combine this switch with /v, DOS displays the number of lines in each file that do not contain the string.
/n	Displays the line number as well as the line that contains the string. You cannot use this switch with /c.
/i	Finds the string with either upper-/ or lowercase letters.

How to use this command You can use this command to find text in a file. The example below looks for the string "Sales" in either upper-/ or lowercase in three files:

```
FIND /i "Sales" FILE_1 FILE_2 FILE_3
```

You can also use the FIND command as part of a filter, as in the following example, which searches for files in the current directory that have a date of November 14, 1990:

```
DIR IFIND "11-14-90"
```

Command type MS-DOS external command with network support.

FOR

What this command does Forms a loop which repeats command that operates on a file several times using a different file each time.

How to enter this command In a batch file:

FOR [*%%c*] IN (*set*) DO *command* [*arguments*]

At the command prompt:

FOR [*%c*] IN (*set*) DO *command* [*arguments*]

How to customize this command

Parameters

%%c or *%c*	Selects the variable that will be used in place of the filename in the command parameter. The c can be any letter between A and Z.
set	Selects the list of filenames, including extensions, that the command will use each time. You can use the * and ? wildcard characters to specify groups of files. The parentheses around the filenames are required.
command	Provides the command that will be performed for each file. It cannot be another FOR command.
arguments	Provides the arguments the command uses. This usually includes the %%c or %c parameter to indicate where the filename should be used.

How to use this command If you wanted to compare several files, you could combine them into one command, as in:

```
FOR %F IN (*.TXT *.DOC *.DAT) DO FC %F A:%F
```

This command compares the files with .TXT, .DOC and .DAT extensions with the versions of these files on drive A.

Command type Batch file command.

FORMAT

What this command does Prepares a disk to store information. You must use this command on floppy diskettes before you can store information on them the first time, or on a hard disk when it is new or when you have changed the partition information.

How to enter this command

FORMAT *drive* [/q] [/v:*label*] [/s] [/b]
FORMAT *drive* [/f:*size*] [/u] [/v:*label*] [/s] [/b]
FORMAT *drive* [/1] [/4] [/u] [/v:*label*] [/s] [/b]
FORMAT *drive* [/8] [/u] [/s] [/b]
FORMAT *drive* [/n:*tracks*] [/t:*sectors*] [/u] [/v:*label*] [/s] [/b]

How to customize this command

Parameter

drive	Selects the drive you want to format.

Switches

/q	Performs a quick format of a disk that erases the file allocation table and the root directory. This switch does not check for bad sectors and cannot be combined with the /f:size, /1, /4, /8, /n:sectors, or /t:tracks switches.
/v:*label*	Adds a volume label to the disk, specified by the text after the colon. If you do not use this switch, the FORMAT command will prompt you for a volume label when the disk finishes formatting.
/s	Copies the three system files to the newly formatted disk. These files are IO.SYS, MSDOS.SYS and COMMAND.COM. The first two are hidden and read-only.
/b	Copies the IO.SYS and MSDOS.SYS hidden and read-only system files to the newly formatted disk.
/f:*size*	Formats a floppy disk to a different capacity than the default for the disk drive. *Size* is the capacity of the disk. This switch can format floppy disks for 160K, 180K, 320K, 360K, 720K, 1.2 MB, 1.44 MB and 2.88 MB, assuming the disk drive is capable of performing such a format. The size can be in any format that represents the size of the disk. For example, a 1.44 MB disk can be entered as 1440, 1440K, 1440 KB, 1.44, 1.44 M, or 1.44 MB.
/u	Unconditionally formats the disk so it cannot later be unformatted by writing over every location on the disk. DOS also performs this type of format when you use /f:size, /1, /4, /8, /n:sectors, and /t:tracks switches.
/1	Formats only one side of the disk (used by 160K and 180K disks only).
/4	Formats a 5-1/4 disk in a high capacity drive to only 360K.
/8	Formats a disk to have only eight sectors on each track, so the 5-1/4 disk is compatible with versions of DOS prior to 2.0.
/n:*sectors*	Selects the number of sectors that the formatted disk will have. This switch cannot be combined with the /f:size switch.
/t:*tracks*	Selects the number of tracks that the formatted disk will have. This switch cannot be combined with the /f:size switch.

How to use this command If you have just opened a box of floppy disks and are ready to use them to store your files, you will need to format each disk before DOS can store the files on it. To format a disk, you would enter the following:

```
FORMAT A:
```

After you press ENTER, DOS prompts for confirmation that the disk is in drive A. When you press ENTER again, DOS will format the disk and then prompt for the volume label. Either type the label and press ENTER or just press ENTER to skip adding a label. After this, DOS displays the disk size information for the disk that has just been formatted.

Notes If the disk is full, you will see a warning that DOS cannot create the file it needs if you later want to unformat the disk. DOS will also display an additional warning if you try formatting a hard disk. If you later realize that you have formatted a disk that you did not want to, you can use the UNFORMAT command to restore as many files from the disk as possible. You cannot use the FORMAT command on drives created by the ASSIGN, JOIN or SUBST command. You also cannot format disks on a network, but you can format floppy disks on drives in a computer connected to a network.

The FORMAT command returns the following exit codes, which you can use in a batch file with IF ERRORLEVEL:

0	Disk formatted successfully.
3	You pressed CTRL-C to stop the command.
4	FORMAT unsuccessfully completed due to an error such as an invalid track 0 or a write protect tab.
5	FORMAT command canceled by providing a negative response to a prompt.

Command type MS-DOS external command.

GOTO

What this command does Redirects execution of commands in a batch file to a different location in the batch file. This commands let you change the order of execution of commands in a batch file. It is often combined with the IF command.

How to enter this command

GOTO *label*

How to customize this command

Parameter

label Specifies a label above the next command that DOS will execute after executing the GOTO command. The label must appear elsewhere in the batch file on its own line, and be preceded by a colon. The label can include spaces, but it cannot include a semicolon, equals sign, or another colon.

How to use this command A batch file that uses the GOTO command might look like this:

```
IF NOT %1==Y GOTO EXIT
CALL MY_FILE
:EXIT
```

If the first variable equals the value of Y, the GOTO command changes the execution to perform the next DOS command after the :EXIT label. If the first variable does not equal the value of Y, DOS performs the MY_FILE.BAT batch file before performing the DOS commands after the :EXIT label.

Command type Batch file command.

GRAFTABL

What this command does Loads the information DOS needs to print extended characters in graphics mode when using a CGA display mode.

How to enter this command

GRAFTABL [*xxx*]
GRAFTABL /status

How to customize this command

Parameter

xxx Selects the code page number to use with GRAFTABL. If this is not provided, DOS uses 437. The choices are: 437 for the United States, 850 for Multilingual, 860 for Portuguese, 863 for French Canadian, 865 for Nordic.

Switch

/status or /sta Displays the active code page loaded with GRAFTABL.

How to use this command If you want to print screens that contain extended characters using the Print Screen key when you are using the CGA graphics display mode, you will want to load GRAFTABL so these characters appear correctly. You would enter GRAFTABL to load the GRAFTABL program, and then when you press CTRL-Print Screen or SHIFT-Print Screen, the extended characters would appear correctly.

Notes GRAFTABL remains in memory, so it uses part of the memory DOS uses for other applications.

Command type MS-DOS external command with network support.

GRAPHICS

What this command does Sets DOS to print graphics screens using the SHIFT-Print Screen key combination.

How to enter this command

GRAPHICS [*type*] [*profile*] [/r] [/b] [/lcd] [/printbox:*id*]

How to customize this command

Parameters

type	Selects the printer you are using to print the graphics screen. You can select:
COLOR1	for an IBM Personal Computer Color printer with a black ribbon
COLOR4	for an IBM Personal Computer Color printer with RGB ribbon
COLOR8	for an IBM Personal Computer Color printer with CMY ribbon
DEFAULT	for any Hewlett Packard PCC printer
DESKJET	for a Hewlett Packard DeskJet printer
GRAPHICS	for an IBM Personal Graphics, Proprinter, or Quietwriter printer
GRAPHICSWIDE	for an IBM Personal Graphics Printer with a wide carriage
LASERJETII	for a Hewlett Packard LaserJet Series II printer
PAINTJET f	for a Hewlett Packard PaintJet printer

QUIETJET	for a Hewlett Packard QuietJet printer
QUIETJET PLUS	for a Hewlett Packard QuietJet PLUSprinter
RUGGED WRITER	for a Rugged Writer printer
RUGGED WRITERWIDE	for a Rugged Writerwide printer
THERMAL	for an IBM PC-convertible Thermal printer
THINKJET	for a Hewlett Packard ThinkJet printer
profile	Selects the name of the file containing the information about the printer selected by type. By default, DOS uses GRAPHICS.PRO in the current directory or a directory accessed by the PATH command.

Switches

/r	Prints black and white as they appear rather than inverting them which is the default.
/b	Prints the background in color when the type is COLOR4 or COLOR8.
/lcd	Prints the graphic screen using the LCD screen ratio rather than the CGA screen ratio.
/Printbox:*id*	Selects the aspect ratio to use. This switch is either /Printbox:lcd or /Printbox:std which must match the text after PRINTBOX for the appropriate printer and display mode in the file containing the printer information.

How to use this command If you want to print a graphics screen using one of the printers selected by the *type* parameter or a compatible, you must use this command before you press SHIFT-Print Screen. An example of using this command is:

```
GRAPHICS DESKJET /r
```

After performing this command, when you display a graphics screen and press SHIFT-Print Screen, DOS sends the image to print in a format the DeskJet printer can understand, and prints white as white and black as black.

Notes GRAPHICS occupies some of the memory used by DOS for other applications. Also, printing graphics screens takes several minutes.

Command type MS-DOS external command with network support.

HELP

What this command does Displays a list of DOS commands and brief description, or displays help inofrmation on specific commands.

How to enter this command

HELP [command]

How to customize this command

Parameters

No parameter entry	Display a list of DOS commands and a brief description of each one
command	Selects the DOS command to display information about

How to use this command Use this command when you want a reminder of the syntax for a command. For example, if you want to see the options avaliable with the DEL command, type HELP DEL and press ENTER to see the following information:

```
Delets one or more files.
DEL  pathname  [/P]
ERASE pathname  [/P]
```

pathname	File(s) to delete. Specify multiple files by using wildcards
/P	Prompt before deleting each file

Note You can also perform this command to display information about specific DOS commands by adding the /? switch to the end of any DOS command. For example, typing **DEL /?** and pressing ENTER will produce the same display shown above.

Command type MS-DOS internal command with network support.

IF

What this command does Evaluates a test, and if the test is true, performs a command. The test can be checking an error level, comparing two strings, or checking if a file exists. The command that this command will perform if the test is

true is often a GOTO or CALL command. If the test is not true, DOS executes the statement immediately following the IF command.

How to enter this command

IF [NOT] ERRORLEVEL *number command*
IF [NOT] *string1 == string2 command*
IF [NOT] EXIST *filename command*

How to customize this command

Parameters

NOT	Negates the value of the test. If the test is true, adding NOT makes it false. If the test is false, adding NOT makes it true.
number	Matches one of the possible exit codes that several programs return. This lets a batch file check whether another DOS command is performed.
string1	Selects the first string to compare. The strings can be characters, with or without quotes; variables, as in %1; or environment variables, as in %DIRCMD%.
string2	Selects the second string to compare. The strings can be characters, with or without quotes; variables, as in %1; or environment variables, as in %DIRCMD%.
filename	Selects a filename whose existance you want to check. You may need to include the drive and directory.
command	Selects the command DOS performs if the test is true.

How to use this command A batch file can use this command in a statement like this:

```
IF %0==Y CALL MY_FILE
```

The IF command tests if the first variable equals the value of Y. If it does, DOS performs the DOS commands stored in the file MY_FILE.BAT and then performs the commands that follow the IF statement. If the first variable does not equal the value of Y, DOS skips calling MY_FILE.BAT and just follows the IF statement.

Command type Batch file command.

INSTALL

What this command does Executes a terminate-and-stay-resident (TSR) program during processing of the CONFIG.SYS file. It is used with the FASTOPEN.EXE, KEYB.COM, NLSFUNC.EXE or SHARE.EXE programs.

How to enter this command

INSTALL=[*drive:*][*path*]*filename* [*parameters*]

How to customize this command

Parameters

drive	Selects the drive containing the program to load, unless it is the current one.
path	Selects the path containing the program to load, unless it is the current one.
filename	Selects the name of the program to load.
parameters	Provides the arguments used by the program. These are unique for each program. You will want to review the other commands for the arguments used by the programs.

How to use this command If you wanted to execute the KEYB.COM program to use the German keyboard when DOS processes the CONFIG.SYS file, for example, you would include a command like this:

```
INSTALL=KEYB.COM GR,850,C:\DOS\KEYBOARD.SYS
```

Command type Used solely in CONFIG.SYS.

JOIN

What this command does Treats the files and directories on the drive that you join as if they are in a directory on another drive.

How to enter this command

JOIN [*drive1*: *drive2:path*]
JOIN [*drive1*] /d

How to customize this command

Parameters

No parameters entry	Displays the drives that are currently joined.
drive1	Selects the drive that you wish to join to a directory on another drive.
drive2	Selects the drive containing the directory that you will use to refer to the joined drive.
path	Specifies the directory on *drive2* that will refer to the files on *drive1*.

Switch

/d	When used with the *drive1* parameter, this switch cancels a previous join statement for the drive.

How to use this command If you wanted to treat the files on drive A as if they were in the TEMP directory of drive C, you would enter the following:

```
JOIN A: C:\TEMP
```

Notes After a drive is joined, you cannot use the original drive specification. You cannot use the directory specified for any purpose other than the JOIN. The BACKUP, CHKDSK, DISKCOMP, DISKCOPY, FDISK, FORMAT, LABEL, RECOVER, RESTORE, and SYS commands cannot be used with a joined drive.

Command type MS-DOS internal command with network support.

KEYB

What this command does Changes the keyboard to use a character set for another language. Use this command to use the keyboard layout of another country's keyboard. Changing the keyboard allows you to access frequently used characters of another language.

How to enter this command From the DOS command prompt or in a batch file:

```
KEYB [xx[,yyy],[pathname] [/e] [/id:nnn]
```

In CONFIG.SYS:

```
INSTALL=KEYB [xx[,yyy],[pathname] [/e] [/id:nnn]
```

How to customize this command

Parameters

No parameter entry	Displays the current keyboard code, the code page for the keyboard code and the active code page.
pathname	Specifies the path and filename of the keyboard definition file, which is normally C:\DOS\KEYBOARD.SYS.
xx	Specifies the two-letter keyboard code.
yyy	Selects the code page to use with the keyboard code.

Switches

/e	Indicates that the keyboard is an enhanced keyboard with two sets of cursor movement keys and twelve function keys.
/id:*nnn*	Specifies the keyboard in use. nnn is a three-letter code.

How to use this command If you were working with a word processing document for another language, such as German, for example, you want to change the keyboard so letters such as the vowels with umlauts (two dots above the letter) can be typed. To make this change, you would enter a command at the DOS prompt or in a batch file that looks like this:

```
KEYB GR,850,C:\DOS\KEYBOARD.SYS
```

This command will change the keyboard from the current setting to the German keyboard. After making this change, the Äand Öwould be to the left of the L, and the ßto the right of the 0. If you wanted to include this command in the CONFIG.SYS file, the command would look like this:

```
INSTALL=KEYB.COM GR,850,C:\DOS\KEYBOARD.SYS
```

Command type MS-DOS external command.

LABEL

What this command does Adds, alters or removes a disk label on a hard or floppy disk. Disk labels can be up to eleven characters that do not include an asterisk, question mark, slash, backslash, vertical bar, period, comma, semicolon, colon, plus sign, hyphen, parenthesis, square bracket, caret, greater than sign, less than sign, or ampersand. Lowercase letters are converted to uppercase.

How to enter this command

```
LABEL [drive:][label]
```

How to customize this command

Parameters

drive	Selects the drive for which you want to add, alter or remove a label if it is not the current drive.

label Provides the label that you want to add to the disk or replace the current label with. If you do not provide a label at the command prompt, DOS will display the current disk label and serial number, and then prompt you for the label. You can type the label and press ENTER or just press ENTER to remove the existing labels.

How to use this command To change the label of drive A to BUDGET FILE, for example, you would enter:

```
LABEL A:BUDGET FILE
```

Notes You cannot use this command with drives that you have created with the SUBST or JOIN commands.

Command type MS-DOS external command.

LASTDRIVE

What this command does Sets the maximum number of logical disk drive letters you can use on your system. This includes drives used with a network as well as logical drives that you use with the SUBST command. Since the table DOS creates to map each potential logical drive letter to the physical drive or logical path it represents, you should choose only as many logical drives as you expect to use.

How to enter this command

```
LASTDRIVE=x
```

How to customize this command

Parameter

x Selects the last letter to use for logical drives.

How to use this command If you want to use several SUBST commands in place of lengthy paths, you will probably need to use the LASTDRIVE command, since DOS assumes that the last logical drive is drive E unless you use the LASTDRIVE command. You might include a command like:

```
LASTDRIVE=H
```

This command provides eight logical drives, A through H.

Command type Used solely in CONFIG.SYS.

LOADHIGH or LH

What this command does Loads a program into reserved memory. Loading a program in reserved memory leaves more room in conventional RAM for other programs. Not all programs can be loaded into reserved memory however.

How to enter this command

LOADHIGH [*drive:*][*path*]*filename*

How to customize this command

Parameters

drive	Selects the drive containing the file that you want to load, if it is not on the current drive.
path	Selects the directory containing the file that you want to load, if it is not in the current directory.
filename	Selects the file that you want to load.

How to use this command To load the file MY_FILE.EXE in reserved memory, for example, you would enter:

```
LOADHIGH MY_FILE.EXE
LH [drive:][path]filename
```

Notes To use this command, you must use the DOS=UMB and DEVICE=EMM386.EXE commands in your CONFIG.SYS file. If reserved memory does not have enough memory to load the program, it is loaded in conventional memory instead.

Command type MS-DOS external command with network support.

MEM

What this command does Displays memory usage and how DOS is using the available conventional, reserved, extended and expanded memory. You can use this command to check the amount of memory available on your system and the memory you have available for storing data and running programs.

How to enter this command

MEM [/program] [/debug]

How to customize this command

Switches

/program	Displays the contents of RAM, indicating the address, name, size and type of every program that is loaded in RAM. Cannot be used with the /debug switch.
/debug	Displays the contents of RAM, indicating the address, name, size and type of every program that is loaded in RAM. Breaks down device drivers into specific devices that are loaded. Cannot be used with the /program switch.

How to use this command This command displays the conventional, reserved, extended and expanded memory that is present and available. You can check the amount of memory on your system by entering:

MEM

Command type MS-DOS external with network support.

MIRROR

What this command does Makes a mirror file of the system; makes a copy of the file partition table; and creates and maintains a delete tracking file. The first two uses allow you to later restore the disk or partition table with the UNFORMAT command, and the last use makes undeleting files more efficient. A mirror file contains information about the file allocation table, boot record and root directory of the selected drive. DOS creates two files: MIRROR.SAV, which is hidden and contains most of the mirror file information, and MIRROR.FIL, which you can see in the directory listing.

How to enter this command

MIRROR [*drive:*] [/t*drive*[*-entries*]] [/1]
MIRROR /partn
MIRROR /u

How to customize this command

Parameter

drive — Selects the size of the delete tracking file by selecting the number of files maintained in this file. The default number of entries depends on the disk's capacity.

Switches

/t*drive* — Selects the drive for which DOS creates a mirror file of the system information. The system information is stored in the MIRROR.SAV (which is also hidden) and MIRROR.FIL (not hidden) files. This parameter can be repeated for each drive for which you want to create a mirror file.

entries — Selects the drive for which DOS creates and maintains a delete tracking file. This delete tracking file is called PCTRACKR.DEL and is created the first time you delete a file, if it does not already exist. This file has a predefined file capacity and size unless you change it with the next switch (entries). Drive is the letter of the drive that you want to track. This switch can be repeated for each drive for which you want to create a delete tracking file. This switch leaves part of the MIRROR command in memory, so it leaves less RAM for other applications.

/1 — Creates a new mirror file without first backing up the MIRROR.SAV file to MIRROR.BAK. The default is to back up this file so you can return to the prior version if the original ever becomes corrupted.

/partn — Creates a copy of the partition table. You must provide the drive letter of the floppy disk where this command will save the partition information in a file called PARTNSAV.FIL. This floppy disk should be in a safe location.

/u — Removes the part of the MIRROR command that remains in memory for updating delete tracking files. This removes deleted file tracking from all drives. This switch will not work if MIRROR was not the last loaded TSR.

How to use this command You can use this command to create mirror files of your hard disks as well as to delete tracking files for these disks. A command like this would be included in the AUTOEXEC.BAT file, so its features would always be available when you use the system. If you had hard disks labeled C and D, you might use this command:

```
MIRROR C: D: /tC /tD
```

Notes Do not use MIRROR to create delete tracking files on drives created with JOIN or SUBST. If you want to use ASSIGN to assign another drive letter to a drive

and use the assigned drive with delete file tracking, use ASSIGN before you use MIRROR.

If you use MIRROR on a disk drive for floppy diskettes, any files you delete using that drive will create a delete tracking file or update the existing one. Files deleted on drives that you have not selected with MIRROR will not update the PCTRACKR.DEL file.

Command type MS-DOS external command with network support.

MKDIR or MD

What this command does Creates a subdirectory using the name you provide. You can create the new subdirectory on the current drive or another drive. Subdirectory names follow the same conventions as filenames although most subdirectory names do not use an extension.

How to enter this command

MKDIR [*drive:*][*path*]
MD [*drive:*][*path*]

How to customize this command

Parameters

drive	Selects the drive that contains the disk where this command will create a directory. If you do not provide it, DOS creates the directory on the current drive.
path	Provides the name of the directory that this command creates. If the path is preceded by a backslash, DOS adds the subdirectory directly beneath the root directory on the drive. If the path does not include a backslash, the new directory will be created beneath the current directory.

How to use this command Use MD to create a logical structure on a hard disk to help you organize your access to files. To create three directories under the current directory, you might type these commands:

```
MD MEMOS
MD LETTERS
MD REPORTS
```

If you use a backslash in front of the subdirectory names, they are added under the root directory rather than under the current directory.

Command type MS-DOS internal command with network support.

MODE

What this command does Configures devices used by your system. You can use it to configure parallel and serial ports, redirect parallel port output to a serial port, set the device code pages, select display mode, set how fast DOS processes repeated keystrokes, and display the current configuration of all devices.

How to enter this command

MODE LPT*#*[:][COLS=*c*][LINES=*l*][RETRY=*r*]
or: MODE LPT*#*[:] [*c*][,[*l*][,*r*]]
MODE COM*#*[:] baud=*b* [parity=*p*] [data=*d*] [stop=*s*] [retry=*r*]
or: MODE COM*#*[:] *b*[,*p*[,*d*[,*s*[,*r*]]]]
MODE LPT*#*[:]=COM*#*[:]
MODE [*device*] [/status]
MODE *device* CODEPAGE PREPARE=((*yyy*[*yyy*])[*pathname*])
or: MODE *device* CP PREP=((*yyy*[*yyy*])[*pathname*])
MODE *device* CODEPAGE SELECT=*yyy*
or: MODE *device* CP SEL=*yyy*
MODE *device* CODEPAGE REFRESH
or: MODE *device* CP REF
MODE *device* CODEPAGE [/status]
MODE *display*,*n*
MODE [*display*],*shift*[,*test*]
MODE CON[:] [cols=*m*] [lines=*n*]
MODE CON[:] [rate=*m*] [delay=*n*]

How to customize this command

Parameters

#	Selects the parallel or serial port you are working with. For parallel ports, this number is 1, 2, or 3. For serial ports, this number is 1, 2, 3, or 4. The colon after the number is optional.
COLS=*s* and *s*	Selects the number of columns that fit in a line. The choices are 80 or 132; the default is 80.

LINES=*l* and *l*	Selects the number of lines that can print in one inch. The choices are 6 or 8; the default is 6.
RETRY=*r* and *r*	Selects the action DOS takes when a timeout error occurs. Choices are: e, to return an error message (which is also the default); b, to return a busy message; r, to return a ready message; or none, to return no signal.
BAUD=*b* and *b*	Selects the baud rate (how many bytes per second are sent through the port). Choices are: 11 for 110, 15 for 150 baud, 30 for 300 baud, 60 for 600 baud, 12 for 1200 baud, 24 for 2400 baud, 48 for 4800 baud, 96 for 9600 baud, 19 for 19200 baud.
PARITY=*p* and *p*	Sets the parity. Choices are: n for none, e for even, o for odd, m for mark, or s for space. The default is even.
DATA=*d* and *d*	Sets the number of data bits. Choices are 5, 6, 7 or 8. The default is 7.
STOP=*s* and *s*	Sets the number of stop bits. The choices are 1, 1.5 or 2. The default is 1 for all baud rates except 110 baud, which uses a default of 2.
device	Selects the device you are using with the MODE command. Choices include: CON (monitor and keyboard), LPT1, LPT2, LPT3, COM1, COM2, COM3, and COM4.
yyy	Selects the code page number that is loaded or selected.
display	Selects the display. Choices are: 40 for forty characters per line, 80 for eighty characters per line, BW for black and white graphics, CO for color graphics, MONO for monocolor display.
n	Sets the number of lines in the display. Choices are 25, 43 and 50, although for the last two, ANSI.SYS must be loaded.
shift	Shifts the display to the right or left. Choices are: R, to shift display one column to the right and L, to shift display one column to the left.
cols=*m*	Sets the number of columns in the display. Choices are 40 and 80.
ines=*n l*	Sets the number of lines in the display. Choices are 25, 43 and 50, although for the last two, ANSI.SYS must be loaded.
rate=*r*	Sets the number of times per second a keystroke is repeated when the key is held down. R can be a number between 1 and 32 to repeat the character 2 to 30 times per second.
delay=*d*	Sets the time that DOS waits between the first and second character when you hold a key down. Choices are 1 through 4 to represent .25/, .5/, .75/ and 1-/ second intervals.

Switch

/status or /sta Displays the status for the device or code page that you select.

How to use this command This command can be used for several reasons. You can use the MODE command to change the settings of a printer attached to a parallel port by entering a command like this:

```
MODE LPT1 132,8,b
```

This command sets the printer connected to the first parallel printer port to print 132 characters across, eight lines per inch and return a busy signal when a time-out error occurs.

You can also use the MODE command to change the settings of a serial port used to connect modems and printers to your computer. A sample use of the MODE command for this purpose would look like this:

```
MODE COM1 30,E,7,1,R
```

This command sets COM1 to use 300 baud with even parity, 7 data bits, 1 stop bit and return a ready message when a time-out error occurs.

The MODE command can also be used to redirect printing from a parallel port to a serial port so printer output is automatically sent to a serial printer. A sample of the MODE command used for this application is MODE LPT1=COM1, which redirects output sent to LPT1 to COM1 (you must make the settings for COM1 before you redirect the printer output).

You can use the MODE command to display the status of one or all devices, either by entering MODE by itself or by entering MODE followed by a device. This returns the settings made with other MODE commands.

You can use MODE to change the code page a device uses by first using a command like:

```
MODE CON CP PREP=((850)C:\DOS\EGA.CPI)
```

This command loads code page 850 to be used with the monitor and keyboard. Once the code page is loaded, you can select the code page by entering the CHCP command or this command:

```
MODE CON CP SEL=850
```

This also requires that you load the NLSFUNC program and that the CONFIG.SYS command includes the COUNTRY and DEVICE commands.

You can change the display mode of the monitor if you have also loaded the ANSI.SYS driver in CONFIG.SYS. An example of this use is:

```
MODE 80,43
```

This example sets the display to 80 columns and 43 lines, which EGA screens can display.

Finally, you can use the MODE command to change the speed with which DOS reacts when you hold down a key to repeat it. An example of this final use is:

```
MODE CON RATE=32 DELAY=1
```

This example sets DOS to realize that you are holding down a key (rather than typing it once) after .25 seconds. Subsequently, DOS repeats the keystroke you are holding down about 30 times each second.

Command type MS-DOS external command with network support.

MORE

What this command does Displays one screen of a file or DOS command output at a time, pausing to wait for a keystroke from the user before displaying the next screen.

How to enter this command

source |MORE
MORE <*source*

How to customize this command

Parameter

source	Selects source of lines to display. This can be a file or another DOS command, such as TYPE.

How to use this command If you had a lengthy text file called MY_FILE and you wanted to see it one screen at a time, you would enter one of the two following examples:

```
MORE <MY_FILE
TYPE MY_FILE IMORE
```

This displays twenty-four lines of the file with --More-- at the bottom. When you press any key, DOS displays the next screen. This continues until you finish viewing the file or press CTRL-C to end the command.

Notes Since this command creates a temporary file while it displays one screen at a time, the current disk must not be write protected and must have room for the temporary file or this command will not work. The temporary file size depends on the size of the source. If you have ANSI.SYS loaded, you can change the number of lines that appear on the screen so that DOS may display 43 or 50 lines at a time.

Command type MS-DOS external command with network support.

NLSFUNC

What this command does Loads the National Language Support Function program, which DOS uses for country conventions and code page switching. It must be used before you can use the CHCP command.

How to enter this command

```
NLSFUNC [pathname]
```

How to customize this command

Parameter

pathname	Selects the file containing the country-specific information. If you do not provide this information, DOS uses COUNTRY.SYS.

How to use this command If you are changing the code page, you would enter a command like this in either a batch file or at the DOS prompt:

```
NLSFUNC
```

After entering this command, you are ready to use CHCP to switch to another code page.

Command type MS-DOS external command with network support.

PATH

What this command does Selects directories that DOS will search for program and batch files when it does not find the files in the current directory.

How to enter this command

PATH [*drive:path1*][;*drive:path2*]...[;*driveD:pathn*]
PATH;

How to customize this command

Parameters

No parameters entered	Displays the existing path.
[*drive:path1*]	Provides the drives and directories DOS will check with each drive and directory, separated by semicolons. These are checked in the order in which they are specified.
;	Causes DOS to begin checking only the current directory after a previous PATH statement that caused other directories to be checked.

How to Use this Command When you want to start a program such as WordPerfect from any directory on your disk and have DOS find the program files that you need, you will want to enter a PATH statement with references to the directories that you need. You might enter:

```
PATH C:\WP;C:\WP\SPELL;C:\;C:\DOS;C:\BAT
```

The first two entries in the preceding PATH statement are added for WordPerfect program files. The other entries are to enable you to find DOS and batch files that you might need.

Notes This command sets the environment variable PATH to the value of the PATH command.

Command type MS-DOS internal command with network support.

PAUSE

What this command does Suspends execution of a batch file and displays a message instructing the user to press a key so the batch file will continue. This command is

usually preceded by commands that display instructions to the user, such as turning the printer on or putting a disk in a drive.

How to enter this command

PAUSE

How To Use This Command

An example of a batch file that uses this command is:

```
@ ECHOPUT disk in drive A and close the drive door
PAUSE
FORMAT A:
```

This batch file displays the message after the ECHO command, and then displays the message "Press any key to continue ..." When the user presses a key, the batch file formats the disk in drive A.

Command type Batch file command.

PRINT

What this command does Prints text files in the background so you can continue doing other tasks. This command is not intended to print data files that you create with other applications, since most applications will have more advanced print features and many data files include special characters that will cause unexpected results when printed by the DOS PRINT command.

How to enter this command

PRINT[/d:*device*][/b:*size*][/u:*value1*][/m:*value2*][/s:*timeslice*][/q:*qsize*][/t][*pathname*][/c][/p]

How to customize this command

Parameter

pathname Selects the file to print.

Switches

All switches except /t, /c and /p must be entered the first time you use the PRINT command.

/d:*device*	Selects the print device where the files are printed. The device must be one of: PRN, LPT1, LPT2, LPT3, COM1, COM2, COM3, or COM4. If this optional argument is omitted, DOS prompts you for the print output device, suggesting PRN as a default.
/b:*size*	Sets the number of bytes in the buffer the PRINT command can use to store part of the file it is printing. The default and minimum is 512 but this can be as high as 16384.
/u:*ticks1*	Selects the number of clock ticks (1/18th of a second) the PRINT command will wait before determining that the printer cannot accept more information. The default is 1, but you can supply a number between 1 and 255.
/m:*ticks2*	Selects the number of clock ticks the PRINT command has to send information to the printer. The default is 2, but you can supply a number between 1 and 255.
/s:*ticks3*	Selects the number of clock ticks the other tasks you are performing have before DOS tries sending more information to the printer. The default is 8, but you can supply a number between 1 and 255.
/q:*qsize*	Selects the number of files that the PRINT command can store in its queue. The default is 10, but you can supply a number between 4 and 32.
/t	Cancels printing the current file, sounds any printer alarm, prints the message "All files canceled by operator," removes all files from the queue, and advances the paper from the current page.
/c	Removes the file specified before the /c switch and any after it (unless it belongs to the /p switch) from the print queue.
/p	Adds the file specified before the /p switch and any after it (unless it belongs to the /c switch) to the print queue.

How to use this command If you wanted to print some text files, you might use this as your first PRINT command; to change the default settings the PRINT command uses:

```
PRINT /d:LPT1 /b:2048 /m:8 /s:20
```

This command sends the output of the PRINT command to LPT1, reserves 2048 bytes of buffer space, and sets DOS to alternate between eight clock ticks for the PRINT command and twenty clock ticks for other tasks you are performing. When you were ready to print a file, such as a batch file, you could use this command:

```
PRINT *.BAT
```

This adds all files in the current directory with a .BAT extension to the print queue, and starts printing the files. If you want to remove one or more files from the queue, you can enter a command like:

```
PRINT MY_FILE.BAT /c YOURFILE.DAT
```

This removes two files from the print queue. You can also remove all files by entering PRINT /t.

Notes Since the PRINT command stores the names of the files it prints rather than temporary copies (as most print spoolers do), you cannot modify the files in the print queue.

While the PRINT command is working, the printer is unavailable for other printing purposes including SHIFT-Print Screen, CTRL-Print Screen and printing from other applications. The print queue reserves 64 characters for the full path information of each file. You may want to use the SUBST command for a drive letter when a path for a file may approach this limit, since SUBST will make the path information shorter.

Command type MS-DOS internal command with network support.

PROMPT

What this command does Changes the appearance of the DOS command prompt that prompts you for the next command you want to perform.

How to enter this command

PROMPT [*text*]

How to customize this command

Parameters

No parameter entry	Returns the command prompt to the drive letter and a greater than sign (>).
text	Provides the text you want to appear in the prompt. You can include these characters:
$q	Displays equals sign (=)

$$	Displays $ sign
$t	Displays time
$d	Displays date
$p	Displays drive and path
$v	Displays DOS version
$n	Displays drive letter
$g	Displays greater than sign (>)
$l	Displays less than sign (<)
$b	Displays vertical bar character
$_	Moves to beginning of next line
$e	Displays ESC character (for ANSI.SYS)
$h	Deletes previous character

How to use this command To change the prompt, DOS displays a prompt to indicate that it is ready to accept another command. To change that prompt, for example, a greater than sign, you would enter this command:

```
PROMPT $P$G
```

Notes This command sets the environment variable PROMPT to the current prompt setting. You can see this environment variable by using the SET command.

Command type MS-DOS internal command with network support.

QBASIC

What this command does Loads the BASIC interpreter that lets you program in BASIC and run the programs. BASIC is a programming language that you can use to develop your own programs. The editor for BASIC programs is similar to the EDIT system editor. Since it is its own programming language, this book does not cover how you can use the BASIC programming language.

How to enter this command

QBASIC[/h][/nohi][/b][/editor][/g][/mbf][/run][*sourcefile*]

How to customize this command

Parameter

sourcefile	Selects the file to load after the QBASIC interpreter is loaded

Switches

/h	Displays the maximum number of lines on the screen that your hardware is capable of displaying. This is usually 25 lines for CGA screens, 43 lines for EGA screens and 50 for VGA screens.
/nohi	Displays the editor of QBASIC without using high intensity so you can see the screen on portable computers and some monochrome screens.
/b	Displays the editor of QBASIC in black and white.
/editor	Loads the DOS editor that the EDIT command loads.
/g	Updates CGA screens faster to eliminate the "snowy" appearance.
/mbf	Converts QBASIC conversion functions that treat IEEE format numbers as Microsoft format numbers.
/run	Runs the loaded file as soon as QBASIC is loaded.

How to use this command If you wanted to load QBASIC and then load and run the CASTLE.BAS BASIC program that accompanies DOS, you would enter the following:

```
QBASIC /RUN CASTLE
```

Once the program runs and you return to the QBASIC editor, you can exit QBASIC by selecting File from the menu bar and selecting Exit.

Command type MS-DOS external command.

RECOVER

What this command does Recovers files from a bad or defective disk. This command reads a disk and recovers sectors from the available files that it can find. Use this command when you have a damaged disk that you need files from. This command may or may not be able to recover your files, so creating backups is a good idea.

How to enter this command

RECOVER *pathname*
RECOVER *drive*

How to customize this command

Parameters

pathname	Specifies the files that you want to recover. This recovers one file, which means that DOS reads as much of the file as it can and puts this information into a file in the root directory called FILE0001.REC (or the next highest number).
drive	Selects the disk from which you want to recover all possible files. This option is for when the entire disk is damaged.

How to use this command This command is used when you have a disk from which you cannot read the data files. To recover all of the files possible from drive A, you would enter:

```
RECOVER A:
```

Notes RECOVER cannot work over a network or on drives created with the JOIN or SUBST command.

Command type MS-DOS external command.

REM

What this command does Adds a line that DOS will not process to a batch or CONFIG.SYS file. After the REM command you can enter any commentary text you want, for example, you can also use this command to temporarily disable a command, the batch file, or CONFIG.SYS.

How to enter this command

REM *comment*

How to customize this command

Parameter

comment	Specifies the text to be included in the REM statement. DOS does not process this text. It must not include the < or > characters.

How to use this command Suppose you have a batch file in which you want to provide explanatory text so another user will understand the commands performed; your batch file might include lines like these:

```
REM Next line checks if end of month processing should be done
IF %1=Y MONTHEND
```

Command type Used by both batch command files and the CONFIG.SYS file.

RENAME or REN

What this command does Changes the name of one or more files. This command cannot change the names of subdirectories.

How to enter this command

RENAME [*drive:*][*path*]*filename1 filename2*

How to customize this command

Parameters

drive	Selects the drive containing the file to rename, if it is not on the current drive.
path	Selects the directory containing the file to rename, if it is not in the current directory.
filename1	Selects the filename to rename. You can use the * and ? wildcard characters in the filename.
filename2	Provides the new filename. You can use the * and ? wildcard characters in the filename. The characters not provided by wildcard characters are the ones that will change in the original filename.

How to use this command You can use this command to rename files. The following example renames all files on drive A that have filenames ending in 89 to end in 90 instead:

```
RENAME A:??????89.* ??????90.*
```

Command type MS-DOS external with network support.

REPLACE

What this command does Replaces all files on one disk with those from another or adds files from one disk to another. This command is used to update the contents of directories.

How to enter this command

REPLACE *pathname* [*drive:*][*path*] [/a] [/p] [/r] [/s] [/w] [/u]

How to customize this command

Parameters

pathname	Selects the source of the files to be replaced or added to another disk.
drive	Selects the target drive where the files are to be replaced or added. If this is not provided, DOS uses the current drive.
path	Selects the target directory where the files are to be replaced or added. If this is not provided, DOS uses the current directory. You cannot specify any filename.

Switches

/a	Adds those files from *pathname* that are not already on the target drive. Files that exist in both locations are not changed.
/p	Prompts you for each file to replace before actually replacing. You type Y to replace the file or N to skip replacing the displayed file.
/r	Replaces read-only files as well as files without this attribute.
/s	Replaces files from *pathname* in any subdirectory of the target directory selected by path. You cannot combine this switch with the /a switch.
/w	Prompts and waits for you to insert a disk into the drive before continuing the command. This switch is used in batch files when the batch file must pause for the user to insert the disk.
/u	Only replaces files in the target location if the target files were created before the source files. This lets you limit replacing files to those that have changed.

How to use this command This command is useful for checking that files that you want on multiple systems are present and up-to-date. For example, if you do not have a network, you may have some files that many people use for reference but do not change. When a change is made to these files, you can use these commands to update the files by copying them from a source that is up-to-date to

the target locations that have not yet received the changed or new files. The following example assumes that the source files are on drive A and that files are normally stored in the C:\BUDGET directory. To bring the target C:\BUDGET directory up-to-date, the batch file would contain these two lines:

```
REPLACE A:*.* C:\BUDGET /A
REPLACE A:*.* C:\BUDGET /U
```

Notes This command returns the following exit codes, which you can incorporate into a batch file with the IF ERRORLEVEL command:

0	REPLACE command successfully replaced selected files
2	DOS could not find files selected by *pathname*
3	DOS could not find the directory selected by *pathname*
5	REPLACE could not replace files due to not having access to the files
8	RAM is full, so DOS could not replace files
11	Wrong syntax for the REPLACE command was entered
15	DOS could not find the drive selected by *pathname*

Command type MS-DOS external command with network support.

RESTORE

What this command does Restores files from a backup disk made with the BACKUP command. The switches and parameters of this command let you select the backed up files that are restored.

How to enter this command

RESTORE *drive*: *pathname* [/s] [/p] [/b:*date*] [/a:*date*] [/e:*time*] [/l:*time*] [/m] [/n]

How to customize this command

Parameters

drive	Specifies the disk containing the backup files.
pathname	Selects the files you want to restore. It can be as simple as a drive, as in C:; it can include a specific directory, as in C:\PROGS; or it can select specific files, as in C:\PROGS*.EXE.

Switches

/s	Restores the subdirectories and files in the subdirectories as well as the files selected by *pathname*.
/p	Prompts for permission to restore files that are read-only and files that have been modified since they were backed up.
/b:*date*	Restores files with a date on or before the specified date. The format of *date* must match the format used by the DATE command.
/a:*date*	Restores files with a date on or after the specified date. The format of date must match the format used by the DATE command.
/e:*time*	Restores files with a time on or earlier than the specified time. The format of time must match the format used by the TIME command.
/l:*time*	Restores files with a time on or later than the specified time. The format of time must match the format used by the TIME command.
/m	Restores files that have been modified since the last BACKUP command by restoring only files that have the archive attribute on the drive.
/n	Restores only the files on the backup disk that no longer exist on the source disk.
/d	Displays the files available to restore selected with pathname.

How to use this command You can use this command when you realize that you need one of the files that you backed up with the BACKUP command. If your entire hard disk became corrupted, for instance, and you needed to restore all of the files (except the COMMAND.COM and two hidden system files), you would enter a command like this:

```
RESTORE A: C: /S
```

Usually, you need to restore only one or a group of files, so you would enter a command like this:

```
RESTORE A: C:\PROGS\*.EXE
```

Notes The RESTORE command returns the following exit codes, which you can use in a batch file with IF ERRORLEVEL:

0	Restore completed successfully
1	No files were found to restore

3 You pressed CTRL-C to stop the RESTORE command
4 RESTORE command halted due to an error

Command type MS-DOS external command with network support.

RMDIR or RD

What this command does Removes the specified directory.

How to enter this command

RMDIR [*drive:*]*path*
RD [*drive:*]*path*

How to customize this command

Parameters

drive Selects the drive that contains the directory to remove. If you do not provide it, DOS assumes the current drive.

path Tells DOS the name of the empty directory that you wish to remove.

How to use this command You should remove unneeded directories from your disk. Although an empty directory does not require a significant amount of disk space, a number of entries that you no longer need clutter your display and should be removed. To remove the directories C:\LOTUS\ACCT and D:\TEMP, for example, you would make these entries with another directory as the current directory:

```
RD C:\LOTUS\ACCT
RD D:\TEMP
```

Notes A directory is empty when only the . and .. entries remain in a directory listing. The command will fail if you attempt to use it on the current directory or a directory that is not empty.

Command type MS-DOS internal command with network support.

SET

What this command does Sets or displays environment variables. These variables can be set and used to control batch files and programs.

How to enter this command

SET [*variable*=[*string*]]

How to customize this command

Parameters

No parameter entry	Displays the environment variables and their current values.
variable	Defines the environment variable you want to set or change.
string	Defines the value of the environment variable defined by variable. The string should not be enclosed in quotes. If you do not provide this parameter, DOS clears and removes the environment variable selected by *variable*.

How to use this command If you wanted to set a TEMP variable to the string C:\OUT_FILES, you would enter the following:

```
SET TEMP=C:\OUT_FILES
```

Notes DOS may set four environment variables: If you change the command prompt to another display with the PROMPT command, the PROMPT environment variable will contain the current prompt setting. If you use the APPEND command with the /e switch, DOS sets the APPEND environment variable to the current APPEND directories. If you use the DIR command, DOS sets the DIRCMD environment variable to the default parameters and switches for the DIR command. DOS sets the PATH environment variable to the path entries you have entered with the PATH command. The COMSPEC environment variable is set to the file path of the COMMAND.COM file.

If DOS does not have enough room to store the environment variables, it displays a message when it runs out of room. You can use SHELL in CONFIG.SYS or COMMAND to reserve room for more environment variables. To use an environment variable in a batch file, enclose the environment variable in %, as in %PROMPT%.

Command type MS-DOS external command with network support.

SETVER

What this command does Displays or sets the version table. The version table sets the version number the VER command returns when specific programs are run, instead of returning the default of 5.00. Some programs require a specific version

of DOS, so this command alters the output the VER command returns when called by the program in the version table. This does not affect other times that you use the VER command.

How to enter this command

SETVER [*drive:*] [*filename*] [*n.nn*] [/delete]

How to customize this command

Parameters

drive	Selects the drive that contains the MSDOS.SYS file. This file is a hidden system file that is often in the root directory of the C hard drive. If you do not provide this information, DOS assumes that this file is on the current drive.
filename	Provides the filename that you want to add or delete from the version table. It does not have to be in the current directory or on the current drive.
n.nn	Selects the DOS version number that should be added to the version table. n.nn the version number as 3.40 or 4.00. This parameter cannot be combined with the /delete switch.

Switch

/delete	Deletes the selected filename from the version table. This switch cannot be used with the n.nn parameter.
/quiet	Suppresses any output when you use this switch with /delete.

How to use this command If you are running a program that was created specifically for DOS 3.3 and you want to run it using DOS 5.0, you may need to add the program name and the DOS version number to the version table. For a program called MY_PROG.EXE, you would enter the following, assuming that you have installed DOS on drive C so that the MSDOS.SYS file is on drive C:

```
SETVER C: MY_PROG.EXE 3.30
```

Notes This command is not covered elsewhere in this book because it is a DOS command that you will only need to run a specific program, and the program's installation program will provide the exact entry you need. Since most programs can run with any version of DOS, you usually will not need to use this command. DOS already provides entries in the version table for many of the popular programs that require a version table entry. Make this change only if you know

that you must make this change to run a program. When you add an entry to the version table, you will see a warning indicating that this may make the program behave unexpectedly. This command has the following exit codes:

0	MSDOS.SYS file successfully changed
1	Used a switch the SETVER command does not use
2	Filename parameter is invalid
3	Not enough memory to perform command
4	N.nn parameter is invalid
5	Did not find filename in version table
6	MSDOS.SYS file not found on selected or current drive
7	Invalid MSSOS.SYS or IBMDOS.COM file
8	Invalid drive provided by command
9	Too many command line parameters
10	MS-DOS version was not provided
11	Missing parameters
12	Error occurred while reading the MSDOS.SYS file
13	Version table in MSDOS.SYS file is corrupt
14	Files do not support a version table
15	Not enough room for a new entry in MSDOS.SYS version table
16	Error occurred while writing the MSDOS.SYS file

Command type MS-DOS external command with network support.

SHARE

What this command does Installs the file sharing and locking capabilities needed when multiple applications may try using the same file at once. This is the case with networks and multitasking.

How to enter this command

SHARE [/f:*space*] [/l:*locks*]

or

INSTALL=SHARE [/f:*space*] [/l:*locks*]

How to customize this command

Switches

/f:*space* Sets the number of bytes reserved for file sharing information. Space is the number of bytes DOS should use. If you do not use this switch, DOS uses the default of 2048 bytes.

/l:*locks* Sets the number of files that may be locked at one time. *Locks* is the number of files you want to lock. If you do not use this switch, DOS uses the default of locking as many as twenty files.

How to use this command This command can be included in your AUTOEXEC.BAT file or CONFIG.SYS file, or entered directly at the DOS command prompt. To include it in the CONFIG.SYS file using the default settings, you would enter:

```
INSTALL=SHARE
```

To enter it in the AUTOEXEC.BAT file or from the DOS command prompt, you would enter: SHARE.

Notes This command remains in memory after it is executed, which leaves less RAM for other applications.

Command type MS-DOS external command with network support.

SHELL

What this command does Executes a command processor and passes any parameters to it. You can use this command to execute a command processor other than COMMAND.COM, or to execute COMMAND.COM in a different way, such as increasing the room reserved for environment variables.

How to enter this command

SHELL=[*drive*:][*path*]*filename* [*parameters*]

How to customize this command

Parameters

drive Selects the drive containing the command processor, unless it is the current one or is specified in the PATH statement.

path	Selects the path containing the command processor unless it is the current one or is specified in the PATH statement.
filename	Provides the name of the command processor.
parameters	Provides the arguments used by the command processor. These are unique for each command processor.

How to use this command If you want to use COMMAND.COM with 256 bytes reserved for environment variables (the default reserves only 160 bytes), you would include this line in your CONFIG.SYS file:

```
SHELL=COMMAND.COM /P /E:256
```

The /P and /E parameters are parameters of the COMMAND.COM command. The /P tells DOS that this command processor is permanent, so it displays a message about the command processor and executes any AUTOEXEC.BAT file. The /E: followed by a number reserves room for environment variables.

Command type Used solely in the CONFIG.SYS file.

SHIFT

What this command does Shifts the value assigned to a variable to the previous variable name. The second through eleventh values will be assigned to variables %0 through %9 instead of the first through tenth. The first will be dropped. This command can be repeated several times to continuously shift values to previous variables.

How to enter this command

```
SHIFT
```

How to use this command A batch file that uses this command might look like this:

```
:TOP
COPY %1 /V A:
SHIFT
IF EXIST %0 GOTO TOP
```

When this batch file is executed, DOS copies the first file selected and then shifts all of the variables. The first time this is done, the first value after the batch file is ignored and the second value is used for the %0 variable. The IF command

checks that this file exists. If the file exists, DOS goes to the TOP label and repeats these commands for the next variable. If the file does not exist, which happens when all of the variables are used, the batch file ends. If this batch file were called MANYCOPY, you could execute this for several files by making an entry like:

```
MANYCOPY  FILE_1.EXE FILE_2.BAT FILE_3.DAT
```

Command type Batch file command.

SORT

What this command does Sorts lines of input, based on the characters in a column. If the selected column for two or more lines contains the same character, this command moves to the next column to determine the order of those particular lines within the large order. The output for this commands is sent to the screen unless you direct it to another device or file.

How to enter this command

[*source*] ISORT [/r] [/+*n*]
SORT [/r] [/+*n*] <[*source*]

How to customize this command

Parameter

source	Selects source of lines to sort. This can be a file or the output of another DOS command such as TYPE. If you do not supply this information, DOS assumes the source information will come from the keyboard.

Switches

/r	Sorts the lines in reverse order instead of the default of ascending order.
/+*n*	Selects the column to use to sort. n is the column number that you want to use to sort. If you omit this switch, DOS uses the first column.

How to use this command If you wanted to sort the current directory according to the sizes of the files, you would make the following entry:

```
DIR ISORT /+13
```

Notes This would sort the files in the directory according to the character found in the 13th column, which, in a directory listing, is the file size. SORT can sort files as large as 64K. The order in which DOS sorts alphabetical letters depends on the country selected with the COUNTRY command, since DOS changes the sort order so international characters are sorted according to their proper alphabetical order.

Command type MS-DOS external command.

STACKS

What this command does Sets the RAM DOS reserves for processing hardware interrupts. Hardware interrupts are signals from hardware, such as pressing CTRL-C on the keyboard or turning off the printer. When DOS receives multiple hardware interrupts, it puts the information for each unprocessed one on a stack.

How to enter this command

STACK=*n,s*

How to customize this command

Parameters

n	Selects the number of stacks, between 8 and 64.
s	Selects the size of each stack, between 32 and 512.

How to use this command If the space reserved for processing hardware interrupts is insufficient, the system may lock up. The default for PCs, XTs and PC-Portables is to not reserve space for stacks. On other machines, such as ATs and PS/2s, the default is nine stacks of 128 bytes each. You can select different stack information by entering a command in the CONFIG.SYS file such as:

```
STACKS 16,64
```

This command creates sixteen stacks of 64 bytes each.

Command type Used solely in CONFIG.SYS.

SUBST

What this command does Associates a path with a drive letter, so you can use the associated drive in place of the path information.

How to enter this command

SUBST
SUBST *drive1*: [*drive2*:]*path*
SUBST *drive1*: /d

How to customize this command

Parameters

No parameter entry	Lists all substituted drives.
drive1	Specifies the drive that will represent another directory.
drive2	Specifies the drive of the directory that you are assigning to *drive1*, if it is not the current drive.
path	Specifies the path that *drive1* will represent.

Switch

/d	Removes the drive assignment created to drive1.

How to use this command Use this command when you do not want to use a lengthy PATH statement. This command provides a shortcut, since you can use a drive such as E: to represent a lengthy path, such as C:\123\BUDGET\1990. For this example you would enter:

```
SUBST E: C:\123\BUDGET\1990
```

After entering this command, you can work with this directory by either specifying the full path or by referring to drive E. You cannot use BACKUP, CHKDSK, DISKCOMP, DISKCOPY, FDISK, FORMAT, LABEL, RECOVER, RESTORE, or SYS with a drive you assign with the SUBST command. When you are finished using the command, enter:

```
SUBST E: /D
```

Notes To use letters higher than E, your CONFIG.SYS file must include a LASTDRIVE command specifying the last letter you expect to use. Once you use this command, you can refer to the path using the substituted letter or the actual path that it represents, unlike JOIN, which will only let you use the path assigned with the command rather than the drive it represents.

Command type MS-DOS external command.

SYS

What this command does Transfers the IO.SYS, MSDOS.SYS, and COMMAND.COM .

How to enter this command

SYS [*drive1:*][*path*] *drive2:*

How to customize this command

Parameters

drive1	Selects the drive containing the system files. If you do not provide this information, DOS uses the current drive.
path	Selects the directory containing the system files. If you do not provide this information, DOS uses the current or root directory.
drive2	Selects the drive where the system files are to be copied.

How to use this command If you have a diskette that already contains files that you want to use as a boot disk, you can use this command to transfer the system files to the diskette. To transfer the system files, assuming the system files are on drive A and you want to transfer them to drive B, you would need to enter the following:

```
SYS A: B:
```

Notes Unlike versions of DOS before DOS 3.3, you no longer need to reformat a disk to add system files to it.

You cannot use this command with drives created with the SUBST or JOIN command. This command also does not work over a network.

Command type MS-DOS external command with network support.

TIME

What this command does Displays or sets the current time in your computer system. You can accept the displayed time by pressing ENTER, or you can type a new time first to change the setting in your computer. DOS will maintain this new setting during the current session. In some newer model computers, this change will remain in effect during future sessions.

How to enter this command

```
TIME [time]
```

How to customize this command

Parameters

No parameter entry	Displays the current time and prompts for the new time. You can accept this time by pressing ENTER or enter a new time using the same format as the displayed time. With times in the afternoon, you can enter the time using a twenty-four hour clock (add 12 to the hours after noon) or using a twelve hour clock, typing a p or PM ater the time.

time	Sets the system time to the time you supply, assuming the time is in a proper format, as described above.

How to use this command If you are using an older model PC when you enter TIME you will see the following response:

```
Current time is  4:23:49.80p
Enter new time:
```

If the current time were 3:24 PM, for example, you would type 3:24p or 15:24 and press ENTER.

Notes Each component of your time entry must be valid. For instance, you cannot enter a time of 2:69 since an hour only has sixty minutes. It is not necessary to enter the number of seconds and hundredths of a second. If you change the country conventions with the COUNTRY command in CONFIG.SYS, the format of the time display and your entry will change to match the country selected.

Command type MS-DOS internal command with network support.

TREE

What this command does Displays the disk directory structure.

How to enter this command

TREE [*drive:*][*path*][/f][/a]

How to customize this command

Parameters

drive	Selects the drive for which you want to display the directory structure.
path	Selects the initial directory that you want to display.
/f	Displays the filenames in each directory.
/a	Uses text rather than graphic characters for the tree structure.

How to use this command To look at an overview structure of another drive, you would enter:

```
TREE D:
```

If you wanted to look at a specific directory on the current drive and see both the subdirectories and files you might enter:

```
TREE \ACCT /f
```

Notes It can be difficult to read the entire directory structure on your screen: You can route the output from this command to your printer with this entry:

```
TREE D:\ACCT /a >PRN
```

Command type MS-DOS internal command with network support.

TYPE

What this command does Displays the contents of a file. This command can also be used with redirection characters to use the file as input to another command, such as MORE.

How to enter this command

TYPE *pathname*

How to customize this command

Parameter

pathname	Selects the file that you want to display. You must include the filename. You can omit the drive and/or directory if the file is on the current drive and/or directory.

How to use this command If you wanted to display the contents of MY_FILE, you would enter the following:

```
TYPE MY_FILE
```

Notes This command only displays the file until it reaches an end-of-file marker. If you want to see the contents of the file beyond the end- of-file marker, use the COPY command with CON as the destination, as in COPY MY_FILE CON.

When you display binary files on the screen, such as graphics and program files, the display will be unintelligible.

Some text files, such as word processing files, contain unusual characters which the word processor uses to record how the text is formatted. These characters will not appear when you use the word processor, but will appear when you display the file with the TYPE command.

Command type MS-DOS external command.

UNDELETE

What this command does Restores one or more files deleted with the DEL or ERASE command. When DOS deletes a file, it changes the first character of the filename to a ? so the other DOS commands will ignore the file. It also marks the data areas the file used as available to store information. UNDELETE can use the information in the file allocation table and directory to restore a file, or it can use a delete tracking file (PCTRACKR.DEL) created by the MIRROR command. Using the delete tracking file produces better results and restores the original first letter of the filename without prompting. Since the area that a file uses is marked as available when it is deleted, it is important that you restore a file with UNDELETE as quickly as you can. Once other files use the space a deleted file originally used, you cannot restore the file.

How to enter this command

UNDELETE [*pathname*] [/list] [/dt] [/dos] [/all]

How to customize this command

Parameter

pathname Selects the filenames to delete. You can provide a single filename, use * and ? to select multiple files, or provide a drive and/or directory information for all files in the drive and directory. If you do not provide *pathname*, DOS assumes that you want to restore all of the deleted files in the directory.

Switches

/list Lists the information for the files that will be undeleted without restoring the files. The information listed is the same that DOS would display if you did not include the /list switch in the command.

/dt Restores files included in the delete tracking file.

/dos Restores files using the information in the file allocation table and directory. This switch causes DOS not to use the delete tracking file.

/all Recovers all deleted files without prompting. DOS uses the delete tracking file if it is available, otherwise it uses the information in the file allocation table and directory. If DOS cannot use the delete tracking file, it substitutes a # for the first character in the filename. If a file exists with that name already, DOS substitutes a %, &, hyphen, number 0-9 or letter A-Z until it has a unique filename in the directory.

How to use this command This command is for immediately (or as soon as possible) restoring a file when you realize that you have accidentally deleted a file that you did not intend to delete. An example of this command is:

```
UNDELETE *.DAT
```

This example searches for a PCTRACKR.DEL file in the root directory of the drive. If this file is found, DOS displays the number of deleted files and breaks these down into files that can be restored, files that cannot be restored and files that can only be partially restored. For each file that can be completely or partially restored, DOS displays the file information and when it was deleted, and then prompts whether you want to restore the file. You can type **Y** to restore the file or **N** to keep the file deleted. This is repeated for every file that you select to UNDELETE.

If the delete tracking file is not found, DOS displays the number of deleted files and breaks these down into files that can be restored and files that cannot be restored. For each file that can be restored, DOS displays the file information and then prompts whether you want to restore the file. You can type **Y** to restore the file or **N** to keep the file deleted.

Next, you must type the first letter of the file. This is repeated for every file that you select to UNDELETE.

Command type MS-DOS external command with network support.

UNFORMAT

What this command does Restores the files and directory on a disk after the FORMAT command is used; restores files in the root directory erased with the DEL command; and recreates damaged partition tables.

How to enter this command

```
UNFORMAT drive:[ /j][ /l][ /test][ /p]
UNFORMAT /partn[ /l][ /p]
```

How to customize this command

Parameter

drive	Selects the drive the REBUILD command will rebuild or the drive to which it will restore the root directory files.

Switches

/j	Tests whether the files created by the MIRROR command agree with the system information on the disk. This tells you whether you can restore the disk from the mirror file without actually rebuilding the disk. It also lets you test whether the mirror file needs to be updated.
/l	Lists the information the UNFORMAT command can restore. If used with *drive*, it lists every file and subdirectory the UNFORMAT command will restore. If used with the /partn switch, it displays the current partition table.
/test	Displays how the REBUILD command will rebuild the information on the disk without using any mirror file. When you use this option the changes are not written to the disk.
/p	Prints the output of this command as well as displaying it on the screen.
/partn	Restores the partition table of a disk. Unless used with the /l switch to display the current partition table, using this switch will prompt for the disk containing the partition file created with MIRROR /partn (PARTNSAV.FIL). After rebuilding the partition table, DOS will reboot the computer. You can then use the UNFORMAT command again to restore the files and directories.
/u	Unformats a disk without using any mirror file.

How to use this command This command is primarily used to restore all of the files that were on a disk before you formatted it. You will not be able to restore the files if you formatted it using the /u switch, or if you have filled the disk with other data. To rebuild an accidentally formatted disk, you would enter:

```
UNFORMAT A:
```

Command type MS-DOS internal command with network support, although you can only use the command on local disk drives.

VER

What this command does Displays the DOS version number.

How to enter this command

```
VER
```

How to use this command You can use this command to quickly check the version of DOS that is installed on a computer. You can use this command by entering **VER**. The result shows that you are using Microsoft's DOS version 5.0.

Command type MS-DOS internal command with network support.

VERIFY

What this command does Sets whether DOS automatically verifies that data is written to a disk correctly by reading the data from the disk after it is written and comparing it to the original. This slows down commands that write to the disk.

How to enter this command

VERIFY [onloff]

How to customize this command

Parameters

No parameter entry	Displays the current VERIFY command setting.
onloff	Turns disk writing verification on or off.

How to use this command This command lets you double check that all information is written to the disk correctly. If you wanted to turn the disk writing verification on, you would enter the following:

```
VERIFY ON
```

Notes Some commands include a switch that lets you verify files as they are written, such as COPY, with the /V switch.

Command type MS-DOS internal command with network support.

VOL

What this command does Displays the disk volume label and serial number.

How to enter this command

VOL [*drive*:]

How to customize this command

Parameter

drive	Selects the drive for which you want to display the volume label and serial number, if it is not the current drive.

How to use this command To display the volume serial number and volume label of the current directory, enter:

```
VOL
```

Notes Disks formatted with versions of DOS before version 4.0 do not have disk serial numbers.

Command type MS-DOS external with network support.

XCOPY

What this command does Copies files (except hidden and system files) as well as subdirectories and the files in these subdirectories. This command includes switches that select which files are copied on the basis of archive attribute, date, or user response.

How to enter this command

XCOPY *pathname1* [*pathname2*] [/a] [/d:*date*] [/e] [/m] [/p] [/s] [/v] [/w]

How to customize this command

Parameters

pathname1	Selects the file or files to copy. Unless you include different information, DOS assumes the files are in the current directory on the current drive.
pathname2	Selects the destination where the files are to be copied. It can include a drive, path information or even new file names if you want the new copy to use different filenames. If this information is omitted, DOS tries copying the files to the current drive and directory using the same filenames.

Switches

/a	Copies only files that have the archive attribute. This switch does not remove the archive attribute from the copied files.
/d:*date*	Copies only files with a date on or after the specified date. The format of date must match the format used by the DATE command.
/e	Copies empty subdirectories if this is combined with the /s switch. Otherwise, XCOPY does not create empty subdirectories on the destination disk.
/m	Copies only files that have the archive attribute, also removes the archive attribute from any copied files.

/p	Prompts for permission to copy each file. You can type a Y to copy the file or N to skip copying the file.
/s	Copies the files in the subdirectories of the source as well as the files in the path provided. If the subdirectories do not exist, this command creates them.
/v	Checks each file after it is copied.
/w	Pauses before copying files and displays a message. You must press a key before XCOPY starts copying files.

How to use this command You can use this command when you want to be able to select which files are copied using only one command instead of several. For example, you might enter:

```
XCOPY *.* A: /P
```

In this example, DOS would prompt for each file, to find out whether you wanted that file copied.

You can also use the XCOPY command to make copies of files that have changed, by entering a command like this:

```
XCOPY *.* A: /M
```

Only the files with an archive attribute are copied, and the archive attribute of the files that are copied is removed. After performing this sample, the only files in the directory that would have an archive bit would be the ones that were modified afterwards.

A third use of the XCOPY command is to copy the subdirectories and their files, as well as the files in the current directory. Suppose you have a disk containing many subdirectories and files and you want to copy both the files, and the subdirectories to your hard disk; your command might look like this:

```
XCOPY A: C: /S /E
```

This example copies the files from the root directory of drive A, creates any subdirectories on drive C to match the subdirectories on drive A if they do not already exist, and copies the files in each of the subdirectories. Since the /E switch is included, if drive A has empty subdirectories, the XCOPY command will create these subdirectories on drive C.

Notes The XCOPY command returns the following exit codes, which you can use in a batch file with IF ERRORLEVEL:

0	All files selected were copied
1	No files were found to copy
2	You pressed CTRL-C to stop the XCOPY command
4	Initialization error due to not enough disk space, or invalid information was provided for the source and destination
5	Disk error occurred while reading or writing the files

Command type MS-DOS external command with network support.

Accelerator keys, 10
Activate Menus (F10) key, 12
Active
    command, reusing, 210-211
    drive, determining, 15
    partition, 123
Add Program dialog box, 103
Advanced dialog box, 106
Advanced option, 106
Advanced Properties option, 105
All Files option, 78
Allocation unit, 117
ALT key, 19, 71-72
    activating menus, 12-13
Ampersand (@) character, with command display, 245
ANSI.SYS file, 316-323
    changing
        modes, 319
        screen appearance, 316-317
    cursor movement codes, 317
    reassigning keys, 320-323
APPEND command, 181-182
    /E switch, 182
    /path:on switch, 182
    /X switch, 182
Application data files, 29
Application programs
    adding to
        new group, 107
        Program Manager, 102-110
    associating file types, 109-110
    changing properties, 106
    copying to other groups, 108
    deleting from
        active task list, 101
        groups, 108
    exiting from, 101-102
    extended memory with, 343
    memory requirements, 105
    printing from, 191
    properties, 103-104
    removing associations, 109-110
    reserved keys, 106
    running from
        Program Manager, 97-98
        Shell, 86-88
    specifying video mode, 106
    switching between, 99-103
Archive attribute, copying files, 156-157
Archive file attribute, 86, 149-152
Arrow cursor, mouse, 12
ASCII text file, 27-28, 241
Asterisk (*) wildcard symbol, 36, 40, 89
ATTRIB command, 7, 59, 149-152
Attribute bit, 149-150
Attributes
    changing, 149-152
    displaying, 150
    file, 86
        setting, 59
AUTOEXEC.BAT file, 9, 250-251, 310-311
    changing international configuration, 323-333
    creating, 310-311
    creating DOS prompt, 300
    freeing up memory, 346-347
    PATH statements, 177-181
AUX device, 284

Background printing, 187-193
BACKUP command, 159-162
    switches, 160-162
Backup dialog box, 112
Backup Fixed Disk option, 111
Backup utility, 111-112
Backups
    adding to existing disks, 160-162
    batch file for, 161
    creating, 159-160
    restoring files, 163-164
Bars
    menu, 66
    scroll, 68
    title, 66
BAT file extension, 28, 87, 241-242
Batch file commands, 260-282
Batch files, 27, 237-257
    backup, 161
    changing sequence of execution, 260-271
    condition tests, 263
    controlling command display, 242-245
    creating with a word processor, 241
    documenting, 249
    DOS commands, entering, 238-239
    editing with Editor, 240-241
    executing, 242
        with CALL command, 274-277
    hard disk backup, 252-253
    interrupting execution, 246-247
    linking, 271-284
    logic errors, 250
    menu system, 253-255
    naming conventions, 238-239
    parameters in, 258-260
        shifting, 281-282
    repeating procedures, 277-279
    running from command-line entry, 104
    syntax errors, 249-250
    testing, 249-250
    understanding, 237-238
Beep, silencing, 92
Bernoulli box, 33
Bookmarks, Editor, 227
Boot disks, creating, 119
Booting DOS, 9-10
Boxes, dialog, 72-75
Buffers, 190
    DOSKEY command, 209
BUFFERS command, 303-306, 338-339
Buttons
    command, 74-75
    option, 75
Byte, 25

CALL command, 274-277
CAPS LOCK key, 71
CD command, 175
Change Attributes command, 86
Change dialog box, 229-230
Character set, setting code pages, 324-328
characters
    - (minus), 68, 80
    + (plus), 68, 80
    (|) Pipe, 297
    + (plus), 155-156
    / (Slash), 16
    > (greater than), 15, 197
    DOS prompt, 300
    ellipsis (...), 72
    redirection, 286-289
    wildcards, 36-40, 82, 89, 136, 152-156
    with filenames, 29-30
CHCP command, 324-327
CHKDSK command, 6, 54-55, 131-133, 338
    /F switch, 132-133, 338
    /V switch, 132
Clicking, mouse, 17
Clipboard, 231-233
CLS command, 20, 244-245
Clusters, disk, 52
Code pages, setting character set, 324-328
Codes
    cursor movement, 317
    exit, 267-269
    formatting, 28
    keyboard, 329-330
    skipping around with GOTO instruction, 260-262
Cold boot, 9
Colors command, 91
Colors, Shell, changing, 91-92
COM device, 284
COM file extension, 26-27, 87
Command
    buffer, DOSKEY, 209
    buttons, 74-75
    editing keys, 210-211
    entry parameters, 104
Command history (F7) key, 208
Command line entries
    adding missing letter, 206
    copying a group of characters, 205

correcting spelling mistakes, 204-205
deleting part of the original entry, 206-207
displaying history, 208-209
maintaining a history, 208-215
recording, 208
repeating with function keys, 203-207
reusing part of the entry, 207
running batch files, 104
saving history, 211
Command output, redirecting, 286-289
Command prompt, 15-17
Command switches, 16
Command syntax, 16
COMMAND.COM file, 3, 9, 119, 308-309
COMP command, 127-128
Computer, start-up, 3
CON device, 284
Condition tests, 263
CONFIG.SYS file, 9, 17, 57, 302-310
changing international configuration, 323-333
commands for, 302-303
device drivers, 311-315
file control blocks, 309
freeing up memory, 346-347
hardware interrupts, 309-310
increasing files/buffers, 303-306
loading
DOS into extended memory, 306-307
programs on startup, 307-308
device driver into reserved memory, 315-316
setting the command processor, 308-309
Confirm on Delete option, 92
Confirm on Mouse Operation option, 92
Confirm on Replace option, 92
Confirmation command, 92
Confirmation dialog box, 92
Context-sensitive help, 19
Editor, 222-225
Conventional memory, 105, 306
Copy (F8) key, 83
Copy command, 43, 59, 61, 83, 152-156, 285-286
+ (plus) with, 155-156
printing with, 193
switches, 155
COPY CON command, 44, 239-240
COUNTRY statement, 330-332
COUNTRY.SYS file, 331
Create Directory command, 89
Create DOS Partition or Logical Drive option, 123
Create Extended DOS Partition option, 123
Create Logical Drive(s) option, 124
Create Primary DOS Partition option, 123
CTRL key, 71
CTTY command, 289-290
Current directory
changing, 175-176
placing subdirectories under, 174
Cursor movement
codes, ANSI.SYS file, 317
keys, 71
Cursors
mouse, 221
selection, 12, 68
Cylinders, disk, 51

Data
copying, 83
moving, 83-84
structure, hard disk, 168-173
DATE command, 20-21
Date entry, copying files, 157
dBASE, file extensions, 32
dBASE IV, temporary exit to DOS, 20
Default devices, 284-285
Default prompt, 299
DEL command, 2, 84, 89, 136
/P switch, 136
wildcards with, 136
DEL key, 89
Delete DOS Partition or Logical Drive option, 122
Delete Extended DOS Partition option, 122
Delete File dialog box, 84
Delete Logical Drive(s) option, 122
Delete Non-DOS Partition option, 122
Delete Primary DOS Partition option, 122
Descending Order option, 82
Deselect All command, 82
Device drivers
ANSI.SYS, 316-323
loading, 312-315
DEVICE command, 312-316, 343-344
DEVICEHIGH command, 315
Devices
changing, 289-290
default, 284-285
HIMEM, 307
specifications, 284-285
understanding, 284-298
Dialog boxes, 72-75
Add Program, 103
Advanced, 196
Backup, 112
Change, 229-230
Confirmation, 92
Delete File, 84
Disk Copy, 111
Display, 235-236
Editor, 222
File Display Options, 74
File Open, 234
Find, 228
Format, 114
Lists, 75
Match Not Found, 229
Move File, 83
New Filename, 234
New Program Object, 102
Rename File, 84
Run, 98
Screen Mode, 75
Search, 89
DIR command, 2, 7, 16, 38-43
switches, 41-43
Directories
- (minus) character with, 68, 80
+ (plus) character with, 68, 80
changing, 79, 175-176
creating, 89
deleting, 89-90
displaying tree, 177
explained, 55
files, listing, 38-39
hard drive, 33
lower-level, 169
parent, 176
physical limits, 168-169
placing subdirectories under, 174
removing, 176-177
reversing default settings, 43
root, 168
selecting files, 81-82
sort order, 42
Directory tree, 68
Disk Copy dialog box, 111
Disk Copy program, 110-111
Disk labels, 130-131
naming conventions, 117, 130-131
Disk Utilities menu, 110-114
DISKCOMP command, 61-62, 126-127
troubleshooting problems, 62
DISKCOPY command, 60-61, 125-126
switches, 60, 126
troubleshooting problems, 61
Disks
adding labels, 131
as file storage media, 26
backup, adding to, 160-162
capacities, 52-55
checking status, 131-133
clusters, 52
comparing, 126-130
comparing file contents, 61-62
configurations, 48-49
copying, 59, 125-126
cylinders, 51
files
erasing, 136
listing, 39-40
moving, 144-146

renaming, 146-147
floppy, 116-120
fragmentation, 336-338
hard, preparing for use, 120-125
improving efficiency, 336-341
retrieving files from damaged, 141-142
sectors, 51-52
source, 125
structure, 49-52
system, formatting, 119
target, 125
tracking information, 136-139
tracks, 50-51
unformatting, 142-143
virtual, 347-349
write-protecting, 57-58, 60
Display dialog box, 235-236
Display Information, command, 54
Display System/Hidden Files option, 82
DOC file extension, 88
DOS
as hardware interface, 3-4
basics, 1-22
changing conventions, 330-332
cold boot, 9
command prompt, 15-17
customizing, 299-334
determining version, 19
different versions, 5
enhancing performance, 349-350
explained, 2
file compatibility with earlier releases, 7
loading into memory, 306-307
main functions, 2-3
new features, 6-7
performance tuning, 335-351
product-specific versions, 6
prompt
changing, 299-301
determining disk capacity, 54
reserved words, 30
returning to Shell, 66
starting/stopping, 8-10
tasks, stopping current, 10
utilities, 4
vs OS/2, 7-8
warm boot, 9-10
DOSKEY command, 7, 208-215
buffer, 209
creating macros, 211-215
DOSSHELL command, 11, 66
/N switch, 93
DOSSHELL.INI file, 91
Double-clicking, mouse, 17
Double greater than (>>) symbol, 287-288
Double-sided/double density disk, 53
Down Arrow key, 19
Dragging, mouse, 17
Drive, changing, 79
DRIVER.SYS file, 312-314
Drives
changing assignments, 311-314
determining active, 15
logical, 121-122
DRIVPARM command, 313
Dual File List option, 77-78

ECHO command, 244-247
EDIT command, 220
/b switch, 221
/G switch, 221
/h switch, 221
/nohi switch, 221
Edit Copy command, 231-232
EDIT.HLP file, 236
Editor, 217-236
bookmarks, 227
Clipboard, 231-233
context-sensitive help, 222-225
customizing, 235-236
dialog boxes, 222
editing batch files, 240-241
exiting, 235
menus, 220-222
screen, 220-221
starting, 218, 220
text editing keys, 226-227
text file
creating, 225
editing, 225-233
printing, 233-235
saving, 233-235
text
finding, 227-229
moving, 231
replacing, 229-230
vs EDLIN, 217-218
EDLIN command, 217-219
Ellipsis (...) characters, with menu items, 72
Enable Switcher option, 100
Encoded file, 188
END key, 19, 71
Environmental variables, 309
displaying, 279-280
removing, 281
Equal sign (=), with commands, 179
ERASE command, 136
/P switch, 136
Error level, testing, 266-269
ERRORLEVEL command, 266
ESC key, 19, 71, 73-74
EXE file extension, 26-27, 87
Executable files, 87
EXIST command, 269-272
EXIT command, 66
Exit codes, 267-269
Exit Shell (F3) key, 11, 19
Expanded memory, 306
using extended memory as, 345
Extended DOS Partition, creating, 123-124
Extended memory, 105, 306, 343
using as expanded memory, 345
Extensions, file, 30-33
External commands, 16-17, 26

Fastopen program, 340-341
FASTOPEN command, 340-341
FASTOPEN.EXE file, 307
FC command, 128-130
switches, 130
FDISK command, 120-122
FDISK menu, 120-123
File Allocation Table (FAT), 37-38
File attributes
changing, 86, 149-152
controlling display, 41-42
displaying, 150
File commands, Shell, 82-90
File control blocks (FCB), 309
File Display option, 82
File Display Options dialog box, 74
File extensions, 30-33
application programs, accepting, 32
associating with a program, 88
BAT, 28, 87, 241-242
COM, 26-27, 87
conventions, 30-32
DOC, 88
EXE, 26-27, 87
table of, 31-32
File list, 68
File locking, 307
File Manager screen, 78-82
File menu, 83-90
File New command, 102, 234
File Open command, 233
File Open dialog box, 234
File Properties command, 106
File types
application data, 29
ASCII text, 27-28
batch, 27
formatted text files, 28
program, 26-27
unformatted text, 28
FILE0001.REC file, 141-142
FILES command, 304-306
Files
adding output to end, 287-288
adding/removing from print queue, 191-192
altering display appearance, 82
ANSI.SYS, 316-323
ASCII text, 241
associating with a program, 109-110
attribute settings, 41-42, 59
AUTOEXEC.BAT, 9, 177-181, 250-251, 310-311, 323-333

backing up, 159-162
batch, 104, 237-257
changing attributes, 149-152
COMMAND.COM, 3, 9, 119, 308-309
comparing, 126-130
contents between disks, 61-62
compatibility with earlier releases, 7
CONFIG.SYS, 9, 17, 57, 302-310, 323-333, 346-347
copying, 152-159
COUNTRY.SYS, 331
displaying, 77-78
with *.* wildcard, 82
DOSSHELL.INI, 91
DRIVER.SYS, 312-314
EDIT.HLP, 236
encoded, 188
erasing from disks, 136
executable, 87
explained, 23-26
FASTOPEN.EXE, 307
FILE0001.REC, 141-142
GRAPHICS.PRO, 196
hidden, 3, 41
HIMEM.SYS, 307, 343
increasing number DOS can open concurrently, 303-306
IO.SYS, 119
joining, 155-156
KEYB.COM, 307
listing, 38-43
MAKECOPY.BAT, 248
managing with Shell, 65-94
marking the copy determination, 157
measuring size, 25
MIRROR.BAK, 137
MIRROR.FIL, 137
MIRROR.SAV, 137
moving/replacing on disk, 144-146
MSDOS.SYS, 119
naming conventions, 29-30, 84-85
NLSFUNC.EXE, 307
opening with Fastopen, 340-341
PARTNSAV.FIL, 139, 144
pausing before copying, 157
PCTRACKR.DEL, 138-139
personalizing naming conventions, 34-36
printing, 85
from background, 187-193
with COPY command, 285-286
program, storing, 27
RAMDRIVE.SYS, 348
redirecting output to, 286-287
renaming, 146-147
representing with parameters, 258-259
restore, 116
restoring from backup, 163-164
retrieving from a damaged disk, 141-142
revising, 37-38
selecting with wildcards, 36-37
SHARE.EXE, 307-308
Shell
changing attributes, 86
copying, 83
deleting, 84
moving, 83-84
renaming, 84-85
searching for, 88-89
selecting, 80-82
SMARTDRIVE.SYS, 339-340
specifying path, 33-34
storage media, 26
system, 41, 119
testing for existence, 269-271
text
creating, 43-44
creating/editing with Editor, 225-233
saving/printing with Editor, 233-235
viewing contents, 44-45
tracking, 136-139
undeleting, 140-141
understanding FAT, 37-38
verifying copy, 126, 154-155
viewing, 40, 85
wide display, 41
Filter commands, 288-296
FIND command, 289-293
Find dialog box, 228
Fixed disks, 50
Floppy disk drive, assigning temporary directory path, 182-184
Floppy disks
adding labels 131
adding system information, 119
checking status, 131-133
copying, 125-126
erasing files, 136
files, renaming, 146-147
formatting, 116-120
for lower capacity drive, 118
moving/replacing files, 144-146
preparing for data storage, 116-120
structure, 49-52
system, starting/stopping DOS, 8-10
tracking information, 136-139
Floppy drive icon, 68
FOR command, 277-279
Foreground, 190
FORMAT command, 116-120, 124-125
Format dialog box, 114
Format option, 113-114
Format utility, 113-114
Formatted text files, 28
Formatting
codes, 28
floppy disks, 116-120
Fragmenting, 336
From text prompt, 83
Function keys, 71
repeating commands, 203-207
Functions, of DOS, 2-3

GOTO command, 260-263
GRAFTABL command, 193-194
GRAPHICS command, 195-197
Graphics display, Shell, 11
Graphics screens, printing problems, 194-197
GRAPHICS.PRO file, 196
Greater than symbol (>), 15, 197
as redirection symbol, 286-287
GW-Basic, 7

hard disk system, starting/stopping DOS, 8-10
Hard disks
adding labels, 131
application-based structure, 169-171
backup batch file, 252-253
checking status, 131-133
data structure, 168-173
departmental structure, 171-172
determining capacity, 54
directories, file storage, 33
erasing files, 136
evolution of, 53
files, renaming, 146-147
formatting, 124-125
individual based structure, 171-173
logical drives, 56-57, 121-122
moving/replacing files, 144-146
organizing, 167-186
partitions, 56, 120-124
preparing for use, 120-125
structure, 49-52
tracking information, 136-139
Hard drive icon, 68
Hardware
code pages, 324
interrupts, setting, 309-310
Help command, 76
Help
(F1) key, 6, 19, 76
accessing, 19, 75-77
in Editor, 222-225
menu, 76, 222
screens, Editor, 222-225
text, adding to programs, 104-105
window, 75-77, 222-224
Hidden file attribute, 86, 149-152

Hidden files, 3, 41
High density disk, 53
HIMEM.SYS file, 307, 343
HOME key, 19, 71

Icons, 68-69
  floppy drive, 68
  hard drive, 68
  Program Manager, 96
IF command, 263-271
Input
  obtaining from another device, 288-299
  redirecting, 286-289
INSTALL command, 308
Installation program, 7
Internal commands, 16-17, 26
International configuration, changing, 323-333
IO.SYS file, 119

JOIN command, 182-184

KB Limit option, 105
KB Required option, 105
Key assignments, Shell, 18-19
Key combinations
  ALT-ESC, 99
  ALT-F (Delete), 108
  Alt-F (File), 13
  Alt-F (Open Program), 97-98
  ALT-F4 (Exit Shell), 11, 19, 66
  ALT-F7 (clear command buffer), 209
  ALT-H (Help), 76, 222
  ALT-O (Options menu), 100
  ALT-TAB, 99
  ALT, to select menu options, 10
  Ctrl-\ (Deselect), 82
  CTRL-/ (Select All), 81
  CTRL-ALT-DEL, 9
  CTRL-BREAK, 10
  CTRL-C (Cancel), 10, 116, 247, 303
  CTRL-END, 19
  CTRL-HOME, 19
  CTRL-INSERT, 232
  CTRL-K (Bookmark), 227
  CTRL-L (Repeat), 229
  CTRL-Print Screen, 188
  CTRL-Q (Search), 228
  CTRL-S (Pause), 247
  CTRL-Z, 239
  Shift-F9 (Exit Shell/w Return), 66
  SHIFT-ALT-ESC, 99
  SHIFT-DELETE, 231
  SHIFT-DOWN ARROW, 81
  SHIFT-ENTER (Add program to task list), 100
  SHIFT-F1 (Help), 222-223
  SHIFT-F8 (Add), 81
  SHIFT-F9 (Exit Shell/w Return), 11, 19
  SHIFT-INSERT, 231-232
  SHIFT-Print Screen, 188, 193-195
  SHIFT-TAB (Previous), 14, 71
  SHIFT-UP ARROW, 81
  Switcher, 101
KEYB command, 329-330
KEYB.COM file, 307
Keyboards
  activating Disk Utilities group, 110
  as device, 285
  codes, 329-330
  deselecting files, 82
  IBM AT Enhanced, 70
  running application programs, 87
  selecting
    files, 80-82
    Shell items, 17, 70-71
  Shell
    changing directory, 80
    changing drive, 79
keys
  accelerator, 10
  ALT, 12-13, 19, 71-72
  CAPS LOCK, 71
  command editing, 210-211
  CTRL, 71
  cursor movement, 71
  DEL, 89
  Down Arrow, 19
  END, 19, 71
  ESC, 19, 71, 73-74
  F1 (Help), 6, 19, 76
  F3 (Exit Shell), 11, 19
  F7 (command history), 208
  F7 (Move), 83
  F8 (Copy), 83
  F10 (activating menus), 12, 19, 72
  function, 71, 203-207
  HOME, 19, 71
  NUM LOCK, 71
  PgDn, 19, 71
  PgUp, 19, 71
  PrtSc (Print Screen), 193
  reassigning with ANSI.SYS, 320-323
  reserved, 106
  SHIFT, 71
  TAB, 12, 14-15, 19, 71
  text editing, 226-227
  UP Arrow, 19

LABEL command, 131
Labels, disk, 117, 130-131
LASTDRIVE command, 57, 311-314
Less than (<) symbol, 288-289
Lists, dialog box, 75
Logic errors, batch file, 250
Logical drives, 56-57, 121-122
  changing assignments, 311-314
  deleting, 122
Loops, avoiding with GOTO statement, 261-262
Lotus 1-2-3
  file extensions, 32
  temporary exit to DOS, 20
Lotus Intel Microsoft Specifications (LIMS), 105
LPT device, 284

Macros
  command line entries, 208
  creating with DOSKEY command, 211-215
  displaying, 213
  parameters, 214-215
  recording, 212-213
  redirecting input/output, 215
  running, 213-214
  saving, 214
Magnetic media, file storage, 26
Main Program group, 14
  adding programs to, 102-103
MAKECOPY.BAT file, 248
Match Not Found dialog box, 229
MD command, 173-174
MEM command, 6, 134-135, 341-342
  /DEBUG Switch, 134-135
  /PROGRAM switch, 134-135
Memory
  clearing when booting system, 9-10
  conventional, 105
  displaying available, 134-135
  expanded, 345
  extended, 105, 343
  freeing up, 345-347
  improving utilization, 341-342
  limiting DOS commands, 345-346
  reserved, 343-345
  setting requirements, 105
  types of, 306-307
Memory resident commands, 346
Menu (F10) key, 72
Menu bar, 66
Menu system, batch file, 253-255
Menus
  activating, 12-13, 72
  Disk Utilities, 110-114
  Editor, 220-222
  FDISK, 120-123
  File, 83-90
  Help, 76, 222
  Options, 78, 82, 90-93
  pull down, 13-14
  selecting items from, 72-73
  Tree, 78
  undoing selections, 73-74
  View, 77-78
Messages
  Bad command or file name, 16-17, 27, 177, 213, 249

Compare OK, 62, 126
Current drive is no longer valid>, 15
displaying/hiding, 92-93
displaying with ECHO command, 246
Duplicate file name or file not found, 147
Errors found, 132-133
File canceled by operator, 192
File cannot be copied onto itself, 153
File creation error, 56
File not found, 16, 32
Files compare OK, 127
General failure reading drive Abort, Retry, Fail?, 116
Insert new diskette for drive, 116
Invalid drive specification, 16, 61
Invalid switch, 16
Not ready reading drive -Abort, Retry, Fail?, 15, 33
Probable non-DOS disk, Continue, 125
Required parameter missing, 16
System will now restart, 121
Minus (-) character, with directory display, 68, 80
MIRROR command, 137-139
MIRROR.BAK file, 137
MIRROR.FIL file, 137
MIRROR.SAV file, 137
MODE command, 197-200, 319
Modes
Shell
determining, 68-69
graphics, 67
text, 67
MORE command, 295-296
Mouse
activating
Disk Utilities group, 110
menus, 13
arrow cursor, 12
changing
directory, 79-80
drive, 79
clicking, 17
copying files, 83
cursor, 221
deselecting files, 82
double-clicking, 17
dragging, 17, 72
moving files, 84
opening programs from Program Manager, 98
pointer, 68
running application programs, 87
selecting
files, 80-82
items in Editor, 221
Shell items, 17-18, 72
to select menu options, 10
Move (F7) key, 83
Move command, 83-84
Move File dialog box, 83
Movement keys, 71
MSDOS.SYS file, 119
Multi-tasking, 5-6

Naming conventions, files, personalizing, 29-30, 34-36
Networks, SHARE.EXE file, 307-308
New Filename dialog box, 234
New Program Object dialog box, 102
NLSFUNC.EXE file, 307
NUL device, 284-285
NUM LOCK key, 71

O switch, DIR command, 42
O: switch, DIR command, 42
On-screen help, accessing, 75-77
Operating system, 2
Option buttons, 75
options
see also commands
Advanced, 106
Advanced Properties, 105
All Files, 78
Backup Fixed Disk, 111
Confirm on Delete, 92
Confirm on Mouse Operation, 92
Confirm on Replace, 92
Create DOS Partition or Logical Drive, 123
Create Extended DOS Partition, 123
Create Logical Drive(s), 124
Create Primary DOS Partition, 123
Delete DOS Partition or Logical Drive, 122
Delete Extended DOS Partition, 122
Delete Logical Drive(s), 122
Delete Non-DOS Partition, 122
Delete Primary DOS Partition, 122
Descending Order, 82
Display System/Hidden Files, 82
Dual File List, 77-78
Enable Switcher, 100
File Display, 82
Format, 113-114
KB Limit, 105
KB Required, 105
Prevent Program Switch, 106
Print, 85
Program/File Lists, 77
Program List, 78
Quick Format, 113-114
Repeat Last Find, 229
Search, 228
Set Active Partition, 123
Shell, file management, 13-14
Show Information, 78
Single File List, 77
Sort By, 82
Options menu, 78, 82, 90-93
OS/2, vs DOS 5.0, 7-8
Output
limiting with MORE command, 295-296
redirecting, 286-289
sorting, 293-295

Parallel ports, 189
changing settings, 198-199
Parallel printer, redirecting output to a serial printer, 200
Parameters
batch file, 258-260
command, 16
macro, 214-215
program command entry, 104
shifting, 281-282
Parent directory, activating, 176
Partitions
active, 123
deleting, 122
displaying information, 121
extended DOS, 123-124
hard disk, 56, 120-124
primary DOS, 123
rebuilding, 144
tracking information, 139
PARTNSAV.FIL file, 139, 144
Passwords, protecting programs, 98
Path, specifying file, 33-34
PATH command, 177-181
Pathnames, substituting drive for directory, 184
PAUSE command, 246
PCTRACKR.DEL file, 138-139
Period (.), with filenames, 29-30
PgDn key, 19, 71
PgUp key, 19, 71
Pipe (|) character, 297
Plus (+) character
joining files, 155-156
with directory display, 68, 80
Pointers, mouse, 68
Ports
parallel, 198-199
printer, 189
serial, 199-200
Prepared code pages, 324
Prevent Program Switch option, 106
Primary DOS partition, creating, 123
Print buffer, customizing size, 190
PRINT command, 85, 187-193, 285-286
Print device, selecting, 189, 191
Print option, 85
Print problems, troubleshooting, 189
Print queue

adding/removing files, 191-192
setting number of files, 191
Print Screen (PrtSc) key, 193
Printers
changing mode, 197-200
ports, 189
sending command output to, 197
using with DOS, 187-202
Printing
files, 85
graphics screens, 194-197
screens, 193-197
text files, 233-235
PRN device, 284
Program/File Lists option, 77
Program files, 26-27
Program groups, 68
adding, 106-107
Disk Utilities, 110-114
opening, 98
reordering items, 108
Program List option, 78
Program Manager
adding program groups, 106-107
associating file types with a program, 109-110
components, 96-99
customizing, 102-110
deleting programs, 108
Disk Utilities group, 110-114
icons, 96
running programs, 97-98
Program properties, 103-104
Programs
see also commands, 110-111
adding
help text, 104-105
to active task list, 100
to Program Manager, 102-110
associating
file extensions with, 88
file types, 109-110
changing properties, 106
command entry parameters, 104
copying to other groups, 108
deleting
from active task list, 101
from groups, 108
Disk Copy, 110-111
exiting from, 101-102
external, 26
Fastopen, 340-341
installation, 7
internal, 26
memory requirements, 105
password protecting, 98-99
removing associations, 109-110
reserved keys, 106
running from Program Manager/Shell, 87-88
SMARTDrive, 339-340
software, 1
specifying video mode, 106
switching between, 99-103
terminate-and-stay resident (TSRs), 138-139, 347
working with, 96-114
PROMPT command, 300-301
Prompts
command, 15-17
DOS, changing, 299-301
From text, 83
To text, 83

Question mark (?) wildcard symbol, 37, 40, 89
Queue, print limitations, 191
Quick Basic, 7
Quick Format option, 113-114

RAMDRIVE.SYS file, 348
Random access memory (RAM), 3, 341-342
RD command, 176-177
Read-only file attribute, 59, 86, 149-152
Read-only memory (ROM), 3
Reading Disk Information box, 9
RECOVER command, 61, 141-142
Redirecting input/output, 286-289
Redirection symbols, 286-289
Releases, DOS, 5
REM command, 249
Rename File dialog box, 84
RENAME command, 146-147
Repeat Last Find option, 229
REPLACE command, 144-146
Reserved memory, 306, 343-345
loading device driver, 315-316
Restore file, 116
Restore utility, 112-113
RESTORE command, 163-164
Root directory, 56
activating, 176
physical limits, 168-169
placing directories under, 173-174
Run command, 87
Run dialog box, 98

Screen mode, Shell, changing, 90-91
Screen Mode dialog box, 75
Screen speed, improving, 341
Screens
clearing, 20
Editor, 220-221
Editor Help, 222-225
File Manager, 78-82
graphics, 194-197
printing, 193-197
Shell, 66-67
View File, 85-86
Scroll bar, 68
Search command, 88-89
Search dialog box, 89
Search option, 228
Search path
changing, 177-181
modifying, 181-182
Sectors, disk, 51-52
Select Across Directories command, 82
Select All command, 81
Selection cursor, 12, 68
Serial ports, 189
changing settings, 199-200
supported, 284
Set Active Partition option, 123
SET command, 279-281
SHARE.EXE file, 307-308
SHELL command, 308-309
Shell
activating from command prompt, 11
application programs, running, 86-88
batch files, executing, 242
beep, silencing, 92
changing interface, 17
collapsing/expanding tree display, 80
colors, changing, 91-92
customizing, 90-93
data, 83
determining disk capacity, 54
determining mode, 68-69
dialog boxes, 72-75
directories
changing, 79-80
creating, 89
deleting, 89-90
directory display, 68, 79-80
drive, changing, 79
executable files, 87
exiting, 19, 66
file commands, 82-90
file list, 68
file management options, 13-14
file management tasks, 65-94
files
altering display appearance, 82
changing attributes, 86
copying, 83
deleting, 84
displaying, 77-78
with *.* wildcard, 82
listing, 38-39
moving, 83-84
naming conventions, 84-85
printing, 85
renaming, 84-85
searching for, 88-89
selecting, 80-82
viewing, 85
floppy drive icon, 68
graphics display, 11

graphics mode, 67
hard drive icon, 68
help system, 75-77
icons, 68-69
interfacing with, 70
key assignments, 18-19
loading, 66
Main Program group, 14
mouse pointer, 68
mouse support, 17-18
overview, 10-15
program group, 68
Program Manager, 96-114
returning to, 66
screen mode, changing, 90-91
screens, 66-67
selecting items, 70-72
starting Editor from, 220
Switcher, 99-103
task list, 68
temporary exit, 66
text display, 12-13
text mode, 67
undoing menu selections, 73-74
view, changing, 77-78
working with programs, 96-114
Shell window, 11-13, 38
default settings, 12-13
moving between components, 15
selected area, 12
selection cursor, 12
Shell Menu (F10) key, 19
SHIFT command, 281-282
SHIFT key, 71
Show Information option, 78
Single File List option, 77
Slash (/) symbol, 16
SMARTDrive, disk caching program, 339-340
SMARTDRIVE.SYS file, 339-340
Software programs, 1
Sort By option, 82
SORT command, 289, 293-295
Source disk, 125
Special characters
DOS prompt, 300
with filenames, 29-30
Specifications, device, 284-285
STACKS command, 309-310
Statements
see also commands
COUNTRY, 330-332
PATH, 177-181
Status line, 67
Strings, testing, 264-266
Subdirectories
- (minus) character with, 68, 80
+ (plus) character with, 68, 80
copying with XCOPY, 158-159
creating, 173-174
explained, 55-56
physical limits, 168-169
SUBST command, 184-185
Switcher
activating, 100
capabilities, 99-103
customizing, 102-110
deleting programs from active task list, 101
disabling, 106
exiting from programs, 101-102
reserved keys, 106
starting application programs, 100
Symbols
- (minus), 68, 80
+ (plus), 68, 80
(|) Pipe, 297
+ (plus), 155-156
/ (Slash), 16
> (greater than), 15, 197
DOS prompt, 300
ellipsis (...), 72
redirection, 286-289
wildcards, 36-40, 82, 89, 136, 152-156
with filenames, 29-30
Syntax, command, 16
Syntax errors, batch file, 249-250
System disks, formatting, 119
System file attribute, 86, 149-152
System files, 41
transferring to a floppy disk, 119
Systems
floppy disk/hard disk, starting/stopping DOS, 8-10
operating, 2

TAB key, 14-15, 19, 71
moving selection cursor, 12
Target disk, 125
Task list, 68
adding programs, 100
deleting programs, 101
running programs from, 87
Terminate-and-stay-resident (TSR) program, 138-139, 347
Text attributes, changing with ANSI.SYS, 317
Text display, Shell, 12-13
Text editing keys, 226-227
Text files
creating, 43-44, 225
editing, 225-233
printing, 233-235
saving, 233-235
viewing contents, 44-45
Text
finding, 227-229
locating with FIND command, 290-293
moving, 231
replacing, 229-230
Text mode, Shell screen, 67
TIME command, 20-21
Time slicing, 190
Title bar, 66
To text box, 83
To text prompt, 83
Tracks, disk, 50-51
TREE command, 7, 177
/A switch, 177
/F switch, 177
Tree display, collapsing/expanding, 80
Tree menu, 78
TYPE command, 44-45

UNDELETE command, 140-141
UNFORMAT command, 7, 117, 142-144
/l switch, 143
/PARTN switch, 144
Unformatted text files, 28
UP Arrow key, 19
Upper memory blocks (UMB), 307
Utilities
see also commands
Backup, 111-112
Disk Copy, 110-111
DOS, 4
Format, 113-114
Restore, 112-113

Variables, environmental, 279-281, 309
VER command, 19
VERIFY command, 126
Video mode, specifying, 106
View, 67
View File Contents command, 85
View File screen, 85-86
View menu, 77-78
View, Shell, changing, 77-78
Virtual disk, 347-349
VOL command, 131
Volume serial number, 118

Warm boot, 9-10
Wildcards
? (question mark), 37, 40, 89
with DEL command, 136
* (asterisk), 36, 40, 89
with DEL command, 136
(*.*) with file display, 82
copying files, 152-156
to specify a filename, 36-37
Windows
Help, 75-77, 222-224
Shell, 11-13, 38
Word processors, creating batch files, 241
WordPerfect
creating unformatted text file, 28
file extensions, 32
temporary exit to DOS, 20
WordStar, creating unformatted text file, 28
Write-protection, disks, 57-58, 60

XCOPY command, 59, 156-159

COPYING A FILE

COPYING A FILE

COPYING A DISK

COPYING A DISK

LISTING FILES

LISTING FILES

CREATING A BATCH FILE

CREATING A BATCH FILE

LOOKING AT A TEXT FILE

LOOKING AT A TEXT FILE

USING THE SHELL FILE UTILITY

USING THE SHELL FILE UTILITY

FORMATTING A DISK

FORMATTING A DISK

CHANGING DIRECTORIES

CHANGING DIRECTORIES

SETTING DATE AND TIME

SETTING DATE AND TIME

NAMING A FILE

NAMING A FILE

USING WILDCARDS

USING WILDCARDS

CREATING A DIRECTORY

CREATING A DIRECTORY

CREATING A BACKUP

CREATING A BACKUP

RESTORING A DISK

RESTORING A DISK

DELETING & UNDELETING FILES

DELETING & UNDELETING FILES

PRINTING A FILE

PRINTING A FILE

CREATING A TEXT FILE

CREATING A TEXT FILE

SORTING A FILE

SORTING A FILE

CREATING CONFIG.SYS

CREATING CONFIG.SYS

USING THE SHELL MENU

USING THE SHELL MENU

IBM/Operating Systems

> $22.95 FPT USA
> $29.95 CANADA

BARNES & NOBLE
B&N $22.95
560-DOS 5.0 AT WORK
4 283788 022952
COMPROG CL 12
0-201-57716-X PI

DOS 5.0 *At Work*

You need a book that makes your work easier, not more difficult. A book that's designed for easy access and simple answers. A book that keeps you out of trouble, but is ready to help if trouble finds you.

DOS 5.0 At Work is the book you need!

Filled with examples and illustrations and written by bestselling author Mary Campbell, **DOS 5.0 At Work** is designed to put DOS to work for you quickly, completely and painlessly.

Here are just a few of the things you'll find in **DOS 5.0 At Work** that you won't find in other DOS books:

- "DOS at Work" sidebars: practical ideas on using DOS from real-world users like you
- "What You'll Learn" objectives at the beginning of each chapter, "At a Glance" summaries, and helpful "Notes" to guide you along the way
- A special "For the Beginner" section written for the first time computer user unfamiliar with the basics
- Specially designed adhesive tabs that allow you to customize the book to your specific needs

Whether you are a beginner or an experienced DOS user, Mary Campbell's **DOS 5.0 At Work** is your essential guide. Written in an easy-to-read and fun-to-use style, it will help you understand the features and capabilities of DOS, solve common problems and pitfalls, and put the power of DOS 5.0 at your fingertips.

Mary Campbell is one of the best-known and bestselling computer book authors writing today. Her books have demystified the world of computers for over half a million readers.

Special Money-Back Guarantee!

We are so confident that **DOS 5.0 At Work** will help you master DOS that we are offering a no-questions-asked money-back guarantee. If for any reason you are not completely satisfied with the book, simply return it within 90 days of purchase, along with your sales receipt and your name, address, and phone number to:

Addison-Wesley Publishing Co., Inc.
Trade Computer Books Division
Route 128, Reading, MA 01867

Your purchase price will be promptly refunded.

52295>
9 780201 577167
ISBN 0-201-57716-X
57716